NASA CONTRACTOR REPORT 166450

Investigation of a Panel Code for Airframe/ Propeller Integration Analyses

S.J. Miley

CONTRACT NAG 2-25
September 1982

NASA

NASA CONTRACTOR REPORT 166450

Investigation of a Panel Code for Airframe/ Propeller Integration Analyses

S.J. Miley
Department of Aerospace Engineering
Texas A&M University
College Station, Texas

Prepared for
Ames Research Center
Under Grant NAG-2-25

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

ABSTRACT

The Hess panel clode was investigated as a procedure to predict the aerodynamic loading associated with propeller slipstream interference on the airframe. The slipstream was modeled as a variable onset flow to the lifting and nonlifting bodies treated by the code. Four sets of experimental data were used for comparisons with the code. The results indicate that the Hess code, in its present form, will give valid solutions for nonuniform onset flows which vary in direction only. The code, presently, gives incorrect solutions for flows with variations in velocity. Modifications to the code to correct this are discussed.

TABLE OF CONTENTS

SYMBOLS

A_{ij}	Influence matrix used in panel code solution
C_L	Lift coefficient
$\overline{C}_L$	Normalized lift coefficient $C_L/C_{L_{MAX}}$
C_T	Propeller thrust coefficient, Reference (6)
C_{T_S}	Propeller thrust coefficient, Reference (3)
$\hat{n}$	Unit normal vector on body surface
ΔP	Difference in static pressure between point on surface and free stream
q_{jet}	Slipstream dynamic pressure
q_∞	Free stream dynamic pressure
V_{jet}	Slipstream velocity
V_∞	Free stream velocity
α	Angle of attack
σ	Panel source density used in panel code solution
σ	Slipstream dynamic pressure function used in Reference (5) $(q_{jet}-q_\infty)/(q_{jet}+q_\infty)$
▮▮▮▮▮▮▮	Width of slipstream

INTRODUCTION

The renewed interest in propeller propulsion systems, stimulated by possible application to future transports, as well as the recognition of aerodynamic problems with existing twin engine aircraft, has created the demand for analytical procedures to treat the combined slipstream/airframe flow field. A number of methods have been developed over the years to determine the flow solution for slipstream/wing combinations. Many of these were created within the last twenty years to support the interest in deflected slipstream short-takeoff and landing vehicles. In all of these methods, however, the wing was treated as a lifting surface, i.e., having no geometric thickness. The effects of thick nonlifting bodies, such as the fuselage and nacelle, were not accounted for. It is well known that the system propulsive efficiency depends on the resultant flow about these bodies, as well as the operation of the propeller itself.

The most practical approach to analyzing the aerodynamic behavior of combined lifting and nonlifting bodies is to use a "panel code". This is a computer program which can calculate, with certain limitations, the flow field and corresponding aerodynamic loads about arbitrary bodies. A study was undertaken to investigate the prediction accuracy of one of these panel codes, the Hess code,[1] compared to existing slipstream/wing aerodynamic test data. The operation of the code and the results of the investigation are reported in the following.

THEORY

Background

The Hess code is a "panel method" computer code which calculates the linear potential flow about arbitrary three-dimensional lifting bodies. The geometry of the various components, which comprise the body, are input as corner coordinates of a set of trapezoidal panels which approximates the surface of the component. Routines[2] are available which calculate these coordinates, given the basic component geometry and a paneling distribution scheme.

Mathematically, the code solves the integral problem

$$\overline{V}_\infty \cdot \hat{n} = -\sigma + \int_S \partial\sigma/\partial n \; ds \qquad (1)$$

When the surface is divided up into panels, on which the respective source densities are assumed to vary in a prescribed manner, the surface integration is performed piecewise on each panel, and equation (1) reduces to

$$\sum_{j=1}^{N} A_{ij} \sigma_j = \hat{n}_i \cdot \overline{V}_\infty \qquad (2)$$

This is an Nth order system of linear equations, for which there are numerous solution algorithms. The A_{ij} influence matrix is formed by determining the normal component of velocity at control point i due to panel j with a unit source density. When there are lifting components, the influence of circulation equivalent unit dipole density is included from the respective lifting panels. After equation (2) is solved, the circulation equivalent dipole strengths are determined through application of the Kutta condition. The term $\overline{V}_\infty$ is called the onset flow and represents the free stream flow seen by

the body. Normally, this term has the magnitude $|\bar{V}_\infty| = 1$. The vector components depend on whether angles of attack and sideslip are specified as part of the input. Equation (2) implies the $\bar{V}_\infty$ is a constant, having the same value for every panel on the body. This is not the case, however. The term $\bar{V}_\infty$ can take on different values for different panels, and accordingly, can be used to represent a shear flow acting on the body. This applies to the case where it is required to represent the effects of a propeller slipstream on the part of the body immersed in the slipstream. Equation (2) can be written in a more general form

$$\sum_{j=1}^{N} A_{ij}\, \sigma_j = \hat{n}_i \cdot \bar{V}_{\infty i} \qquad (3)$$

where $\bar{V}_{\infty i}$ indicates that the onset flow for panel (i) is used. An illustration of this procedure is given in Figure 1. Figure 1(a) shows a paneled representation of a wing and nacelle with the conventional uniform onset flow. In Figure 1(b), the onset flow is varied for different streamwise panel strips to represent a propeller slipstream. Each of the respective onset flow strips in Figure 1(b) represents a $\bar{V}_{\infty i}$ vector in equation (3). The magnitude and angle of attack of the onset flow strip are determined from the axial and swirl flow components of the particular slipstream flow.

Application of the Theory

The purpose of the project was to test the Hess code, in its present form, as a procedure for calculating propeller slipstream induced airloads on fuselage/wing/nacelle configurations. A literature survey was performed to identify applicable experimental data for comparison. Four different documented experiments were selected, and the pertinent geometric and test

condition data were coded for input. The results of the comparison calculations are presented later.

As part of the project, a copy of the Hess code was provided by NASA for the necessary calculations. The code required some modification to make it compatible with the IBM based operating system of the university computer. During this phase, it became evident that gains could be made in reductions of the core storage requirements and execution times. These modifications are described in the following.

CODE MODIFICATIONS

The original version of the Hess code was developed on an IBM 370 computer using the IBM FORTRAN IV language. The version which was made available by NASA for this project, had been modified to include the DERIV input geometry package coded in CDC FORTRAN. DERIV is described in Reference 2. This version was subsequently modified to run on IBM FORTRAN H, Optimization Level 2. The modifications consists of two parts; conversion of the CDC FORTRAN logic to equivalent IBM FORTRAN logic, and compression of core storage requirements and reduction in execution time. An additional modification was subsequently made to facilitate the input of the nonuniform onset flow velocity. This is discussed later in the report. The final version required only 268 kilobytes of memory in an overlay structure, and reduced execution time by a factor of two for a full one-thousand panel case.

The remainder of this section is devoted to a detailed description of the logic and compression modifications made. It is intended for those who wish to make similar modifications. It assumes that the reader is familiar with the program structure and has a pertinent FORTRAN source listing.

CDC Logic Conversion

The modifications to the CDC logic involve four elements. The first of these is the use of a positive-indefinite variable for input array test purposes. The CDC octal code for this value is 1777700000000000000B. The equivalent IBM hexadecimal value is ZF7000000. In terms of real numbers, this code denotes a combination of a zero mantissa with an infinite exponent.

This value cannot be generated arithmetically or through input conversion, therefore it serves ideally as a test variable.

Particular input arrays are loaded with this value through the entry INDEF in subroutine ZERO prior to input. After input, the presence of this value is used to compress the input data in subroutine SQUEEZE prior to output on a mass storage device. It is also used for test purposes in PANELIN. Here, the test variable must be integer. The test is invalid between two real variables because the ZF7000000 code is converted to a floating point zero prior to performing the test. Also, in regard to this test in SQUEEZE and PANELIN, the masking part of the test used in the CDC code, can be removed with no difficulty. A questionable statement occurs in the DO 45 loop in PANELIN in this regard. A conversion is made to a variable KHECK as part of an IF test. This statement does not work, as required, in IBM FORTRAN and must be removed.

The second element concerns the use of the LOCF function throughout the code. Its purpose is to provide the memory address of the argument. Its only essential use is in the subroutine ZERO. Calls to ZERO and INDEF involve a string of memory from a common block whose length is determined by specifying an initial argument and a final argument in conjunction with LOCF. Subroutine ZERO and its associated calls were recoded so that the initial argument and the number of arguments of the common block string are used. A listing of the original and modified ZERO and CALL's are given in Figure 2.

The third element is the conversion from CDC specialized mass storage I/O (input/output) to IBM random access I/O. This involves the subroutine READRS with entry WRITRS. The CDC system includes record keeping which must be explicitly coded for conversion. The OPENMS statement in NUED is replaced

by DEFINE FILE. Also, the CDC I/O functions READMS and WRITMS are replaced by explicit coding in RANDIO. Because of the range of array sizes involved, the modified READERS subroutine stores the smaller arrays in core and writes the rest to mass storage using random access I/O. The storage and record keeping arrays are dimensioned for a maximum of five bodies (nonlifting) and five panels (lifting). The subroutine, its associated calls, and variable initializations are listed in Figure 3.

The fourth element is a coding change in subroutines VFMNLF and VFMLFT necessary to run optimization level 2. This level is sensitive to the placement of ASSIGN and computed GO TO statements. Identical sequences of these are used in both subroutines. The coding was changed to use arithmetic IF statements. The old and modified coding are given in Figure 4.

Core Storage Compression and Execution Time Reduction

Figure 5 shows the overlay structure of the program with original significant common and/or dimension memory requirements in words. The important block is the 20,000 area in the subroutine COLSOL, which is the large A_{ij} normal velocity matrix solver. This 20,000 block is used for the core-resident part of the matrix which is being worked on by the solution algorithm. A large amount of mass storage I/O is required to shuttle pieces of the matrix to and from this block. The compression modification more than doubles the core-resident space for the matrix. This in turn reduces the number of required disk I/O operations. For a one thousand panel solution on the university Amdahl V6 computer, the execution time was reduced by a factor of two. The modification is one of replacing all large memory blocks by a single labeled common block

dimensioned 42,900. The first 16,500 locations are used for labeled common blocks VRONSF, CTABLE, NORMAL, ATABLE, XYZAFT, and STABLE. To make this area available for COLSOL, the first 15,400 words are written on I/O unit 3 prior to the onset flows. The SIGMA array is not required until after COLSOL. The use of unit 3 for temporary storage is based on the desire to reduce the core required I/O buffers. In this regard, units 12 and 15 have been removed from the program. Unit 12 was used for the second half of the source velocity matrix. Most disk packs can hold the entire matrix and there is no gain by using two I/O units. The entire matrix is written on unit 11. Unit 15 was used in COLSOL to store the lower triangular part of the matrix solution for future use. In its present form, the code has no matrix solution for future use. The code also has no provision for solving different right-hand-sides on separate runs, and hence, there is no requirement for maintaining this data on unit 15. Returning now to the temporary storage of the 15,400 words on unit 3, this is done in VFORM prior to writing the onset flows. For subsequent onset flow reads from unit 3, the data set is positioned by reading the known 15,400 word record back to its location in the common block. After COLSOL has been completed, the 15,400 word block is read back in VELPRS. The data set is then positioned for subsequent onset flow reads. With the use of the large common block, all array equivalencing is eliminated to take advantage of the optimizer. Care must be taken in PKUTTA where the logic requires PP and DELB to be equivalenced, or one substituted for the other. This is also the case in VELPRS for B and DD. The modified coding relating to the use of the large common block is given in Figure 6.

TEST CASES

Four test cases were run. The requirements for each case were published data of experimentally determined spanwise lift distributions of a wing immersed in a slipstream flow. In two of the cases, the lift distributions were measured using spanwise wing segments with individual force transducers. In the other two cases, surface static pressure measurements were used. The published description of the test conditions varied from inadequate to fully adequate. The test cases and resulting comparisons with the Hess code are discussed in the following.

Panel Modeling and Program Input

The DERIV geometry routines were used for input and panel modeling. A general panel modeling scheme was applied to all of the test cases. The scheme was based on the 1100 maximum panel limit by the version of the code used.

The propeller afterbodies, when present, were modeled as a duodecogonal cylinder. The cylinders were faired into the wing using straight-line taper in the vertical (Z) direction only. The afterbody chordwise panel density was made similar to that of the wing where the geometries overlapped. The entire spanwise section of the model occupied by the afterbody was assumed as nonlifting.

The wing components were paneled according to the spanwise lift distribution measurement scheme of the respective test models. Where individual force measurement wing segements were used, the spanwise panel boundaries coincided with the segment boundaries. Where chordwise pressure distribution measurements were used, the spanwise panel boundaries were selected so that the panel control points coincided with the spanwise

location of the pressure port rows. The chordwise panels were located according to the cos/sinh distribution function available in the DERIV procedure. Where possible, twenty chordwise panels on each of the upper and lower wing surfaces were used. Previous experience with the code indicated that this number was sufficient to give chordwise pressure distribution results of acceptable detail and accuracy.

A modification to the code was made to improve the input of the nonuniform onset flow. In the original version, a delta velocity vector is input for each panel, once for the entire test case run. The final onset flow vector array used,is the vector sum of the uniform onset vector array (magnitude=1) and the input delta velocity vetor array. This means that when a particular nonuniform onset flow is desired, a delta input flow must be calculated which will produce this desired flow within the program. A second problem is that only one onset flow could be entered for each run. The modification solved these problems by shifting the delta velocity input to the only point in the program where the values are required. This point is in subroutine VFORM. Here, the final onset flows for the matrix solution are being formed for each of the input angles of attack. The input of delta velocity at this point allows a different nonuniform onset flow for each angle of attack, i.e. multiple nonuniform flow inputs per run. Also, the coding is altered so that the actual desired nonuniform flow is input and used as such, instead of the previous intermediate delta flow. To avoid the inputting of a set of values for each of the 1000+ panels, the first and last index numbers for all of the panels in a particular streamwise strip are input with the respective nonuniform velocity vector values for that strip. This is the model shown in Figure 1(b).

In three of the four test cases, usable slipstream data were available. In the fourth case, the slipstream was calculated from given propeller geometric and operating data. The slipstream was input to the program as a set of nonuniform flow vectors as shown in Figure 1(b). The vectors were parallel to the chordwise plane, having an angle of attack only. For the calculated slipstream case, contraction effects were ignored because of the relatively low loading of the propellers. For the available slipstream data cases, the data were taken well downstream of the propeller plane where contraction is of no consequence. Accordingly, the slipstream input velocity vectors had no sideslip angle component.

Test Case I, Reference (3)

This test case was taken from Reference (3). The test model, shown in Figure 7, is a reflection plane configuration consisting of fuselage, wing and nacelle. The slipstream flow is generated by propellers. The wing was divided into eight segments, each providing direct force measurements of lift, drag and pitching moment. Two propellers having different pitch and chord distributions were used. The slipstream characteristics for the test conditions were inadequately described. There were sufficient propeller blade geometry and operating data to calculate the theoretical slipstream characteristics.

The propeller performance method of Reference (4) was used for this purpose. As part of the solution procedure, the method determines the radial gradients of thrust and torque. These values were calculated for the conditions described in the reference, and then used to compute the

axial and swirl velocity components in the far wake by classic momentum analysis.

The analytical model was paneled using seventeen (17) uniform spanwise strips. Each strip contained twenty (20) chordwise panels on the upper and lower surfaces respectively. The fuselage body was ignored, and replaced by a wing panel. The nacelle was paneled as a twelve (12) sided polygon with the longitudinal paneling concentrated near the leading edge of the wing. The nacelle paneling increased in length toward the propeller and toward the trailing edge from this point. The complete spanwise section of the wing occupied by the nacelle was treated as nonlifting. The model paneling is shown in Figure 8.

The test conditions calculated were for wing angles of attack at 0° and 10°. The propeller operating condition was specified in the reference as a thrust coefficient defined in terms of disk loading and slipstream dynamic pressure. A mid range value was selected for the slipstream characteristics. The comparative data were taken from Figures 24 and 42 in Reference (3). The comparisons are shown in Figure 9. The trends evident in the figure hold for the remaining test cases. The code generally overpredicted the lift magnitudes while giving lift distributions similar to the experimental. The errors in the lift magnitudes increased with test condition angle of attack. This is not unexpected as the code does not account for loss of lift due to separation and stall. The discrepancies in lift distribution are due in part to the use of calculated slipstream characteristics rather than actual measured data.

Test Case II, Reference (5)

The test model is shown in Figure 10. The configuration is of an infinite aspect ratio wing. The slipstream is produced by a propeller with drive system located upstream. The lift distribution is determined by direct force measurements from the individual spanwise segments. The characteristics of the slipstream are well documented in terms of both axial and swirl components.

The mathematical model was paneled with twenty (20) spanwise segments. Each segment contained twenty (20) chordwise panels on the upper and lower surfaces respectively. The spanwise paneling distribution was the same as the experimental model except that the narrow width strips were continued across the slipstream through the spanwise wing center. The model panel layout is given in Figure 11.

Six different test conditions were calculated. All were at an angle of attack of $6\frac{1}{2}^{\circ}$, and included a no-slipstream condition. The spanwise lift coefficient distribution used in the reference was based on a function of both the free stream and the slipstream dynamic pressures. No correcting factor could be developed from the reference which would adjust the calculated results for this. Therefore, a normalized lift coefficient was used for comparison. All experimental data were normalized by the maximum value of the experimental lift coefficient for the no-slipstream flow case. The calculated data were correspondingly normalized by the calculated no-slipstream maximum lift coefficient. Figure 12(a) shows the no-slipstream case in terms of the respective normalized lift coefficients. Figures 12(b) and 12(c) give the comparisons for two of the five remaining test conditions. The σ values listed are the aforementioned dynamic pressure functions.

Increasing σ indicate increased slipstream velocities. The computer code again is approximately modeling the distribution but overpredicts the magnitude within the slipstream. The overprediction increased with increasing slipstream velocity.

Test Case III, Reference (6)

The experimental model, shown in Figure 13, is similar to the model of Test Case I. It is a reflection plane model consisting of a wing, nacelle and propeller. The propeller span extends beyond the wing tip. The lift distribution was measured by surface static pressure distribution. The slipstream characteristics were adequately described in terms of both the axial and swirl components. The wing section was symmetrical and the test conditions included negative as well as positive angles of attack. Different propeller thrust levels as well as no-slipstream were run.

The mathematical model was paneled so that the spanwise segment centers aligned with the chordwise rows of static pressure ports. Because of the number of panels used, the chordwise paneling was reduced to seventeen (17) instead of twenty (20). The nacelle was paneled similar to Test Case I. The model paneling is shown in Figure 14.

The comparison results are given in Figure 15. Figure 15(a) is the no-slipstream condition. Here, the code underpredicts the lift magnitudes. This underprediction also occurred with test case IV, and is discussed there. In Figure 15(b), at a low thrust coefficient, there is good agreement. However, in Figure 15(c) at a higher thrust coefficient, the magnitude overprediction occurs again. The distribution patterns are predicted quite well.

Test Case IV, Reference (7)

The experimental model is an infinite aspect ratio wing as in Test Case II. The model is a wing only. The slipstream was created by an axial flow blower located upstream of the wing, Figure 16. The slipstream contained only an axial component. There is no angle of attack variation due to swirl. The lift distribution was determined from surface static pressure ports. The test conditions consisted of variations in the axial flow velocity.

The analytical model was paneled so that the spanwise segment centers aligned with the chordwise pressure port rows. Twenty (20) spanwise strips and twenty (20) chordwise panels were used. The model is shown in Figure 17. Only the spanwise panel boundaries are shown. An illustration including the chordwise boundaries was not of sufficient clarity due to the high line density.

The comparison results are shown in Figure 18. Again the code predicts higher magnitudes in the slipstream region. Outside of the slipstream, the code underpredicts the lift distribution. As with the propeller tests, the prediction error in the slipstream region increases with increasing slipstream velocity.

Figures 19 and 20 give experimental and calculated chordwise pressure distributions respectively for spanwise locations inside and outside of the slipstream. The data are for the flow condition of Figure 18(b). It is repeated here that this test case is for a jet flow, which has no angle of attack variation, unlike the previous propeller test cases. Figure 19 shows the expected increase in pressure magnitude within the slipstream. It is noted that the pressure approaches free stream static at the trailing edge

regardless of the existence of the slipstream. The corresponding calculated results are given in Figure 20. There are two distinct variations from the experimental. First the slipstream distribution does not return to free stream at the trailing edge. A comparison of the two distributions in Figure 20 showed that the calculated airfoil chordwise velocity distribution within the slipstream was an exact magnification of the velocity distribution outside of the slipstream by a factor of the jet velocity ratio ($V_{jet}/V_{\infty} = 2.5$). The effect of the nonuniform onset flow input was that of a pure magnification. The calculated pressure distributions result from the set of velocities that are the solution of Hess code. Pressures are not dealt with specifically. If the inside slipstream distribution were corrected by subtracting the total pressure change due to the jet velocity, the trailing edge pressure would fall well below free stream static. In addition, there would still be a large difference at the leading edge between the calculated and experimental distributions. The conclusion offered is that the code does not calculate the correct lifting body chordwise distribution for nonuniform onset flows. There is a circulation change occurring experimentally that is not accounted for as the code is presently formulated.

The second variation between experiment and calculation is in the airfoil chordwise distributions outside of the slipstream. These are shown in Figure 21. There is considerably more disagreement for a pure airfoil problem than has been previously experienced by the author. Figure 21 shows the source of the underprediction seen in Figure 18. A possible explanation lies with the paneling of the airfoil used in the test. The airfoil is a NACA 0009, which has a smaller leading edge radius than the other airfoils used in the test cases. It is possible that the panel density should have

been increased in the leading edge region to obtain a better representation of the rapidly varying noise curvature. For the underprediction in Test Case III, seventeen (17) panels were used for the airfoil instead of the selected twenty (20). Here again it is possible that the airfoil was under paneled for the desired accuracy. In both cases, the 1100 panel limit did not allow increases in the paneling, and the question of the cause of these variations is still unresolved.

SUMMARY

Comparisons with the test cases show that the Hess code gives acceptable representations of the variation in lift induced by a propeller slipstream. The code tends to predict higher lift magnitudes than measured for increased angles of attack and slipstream velocities. These errors are due to the lack of a formulation within the code to treat the problem of a lifting body with jet flow. The code appears to give valid results for nonuniform onset flows where only the angle of attack varies, i.e. the velocity magnitude remains unity.

If a modification to the Hess code can be made which will work the jet flow solution, then the code will be suitable for the slipstream/airframe problem. A possible solution to this problem is to utilize singularity distributions to model the slipstream instead of the onset flow. The strength and distribution functions of the singularities are determined beforehand to produce the desired slipstream characteristics of the isolated propeller. These known singularities are then included into the Hess code formulations for the induced velocity influence coefficients. The nature and geometric location of these singularities is only partly evident at this time. The axial flow component can be represented by a radial distribution of ring sources at the propeller disk. This is equivalent to semi-infinite co-axial vortex tubes which is the classic means of representing the axial flow of the propeller wake. The swirl component is considerably more difficult. The singularity surfaces, which generate this flow, will penetrate the body geometry and result in numerical difficulties when determining the induced flow

components. The modeling of this flow can be done either with vortices or doublets; however, there is insufficient experimental data to provide proper guidance in this matter.

In summary, the results of this investigation support the use of Hess code to study propeller propulsion system integration problems. Care must be exercised at this stage when considering the validity of the code's predictions for high lift and high thrust conditions. Improvements in the prediction accuracy seem possible by replacing the variable onset flow with singularity distributions. This, however, will require an experimental program to determine the most practical modeling of the slipstream swirl component.

REFERENCES

1. Hess, J.L., "Calculation of Potential Flow about Arbitrary Three-Dimensional Lifting Bodies - Final Report," McDonnell Douglas Corp., Report MDC J5679-01, October 1972, from NTIS as AD-755 480.

2. Halsey, N.D., "A Three-Dimensional Potential-Flow Program with a Geometry Package for Input Data Generation," NASA CR 145311, March 1978.

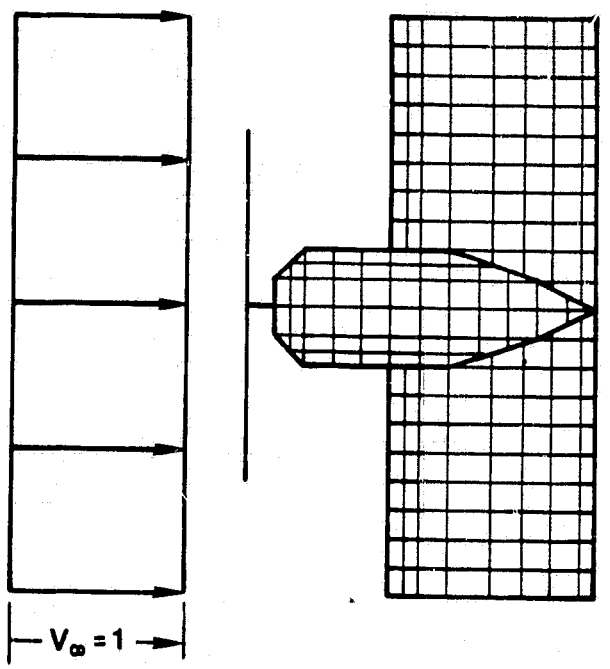
3. George, M. and Kisielowski, E., "Investigation of Propeller Slipstream Effects on Wing Performance," USAAVLABS Technical Report 67-67, November 1967.

4. Pauling, D.V., "The Effects of Uncertainties on Predicting Rotor and Propeller Performance," MS Thesis, Department of Aerospace Engineering, The Pennsylvania State University, May 1975.

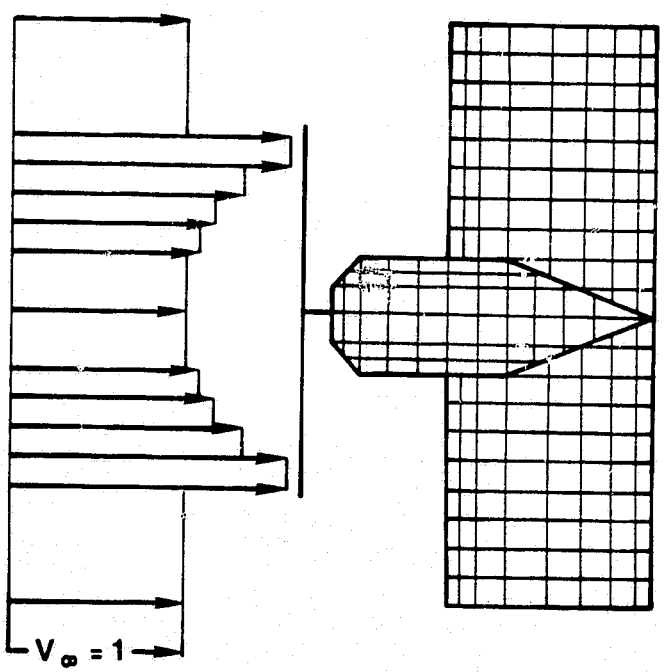
5. Brenckmann, M., "Experimental Investigation of the Aerodynamics of a Wing in a Slipstream," Technical Note No. 11, Institute of Aerophysics, University of Toronto, April 1957.

6. Dunsby, J.A., Currie, M.M. and Wardlaw, R.L., "Pressure Distribution and Force Measurements on a VTOL Tilting Wing-Propeller Model," Report LR-252, National Research Laboratories, Ottawa, Canada, June 1959.

7. Snedeker, R.S., "Experimental Determination of Spanwise Lift Effects on a Wing of Infinite Aspect Ratio Spanning a Circular Jet," Report No. 525, Department of Aeronautical Engineering, Princeton University, February 1961.



(a) Uniform onset flow



(b) Variable onset flow

Figure 1. Use of variable onset flow to represent propeller slipstream.

```
      SUBROUTINE ZERO (A,B)
      DIMENSION A(1)
      STUFF=0.0
10    N=LOCF(B)-LOCF(A)+1
      DO 20 I=1, N
20    A(I)=STUFF.AND.(LOCF(A(I)).OR.7777000000000000000B)
      RETURN
      ENTRY INDEF
      STUFF=1777000000000777777B
      GO TO 10
      END
```

(a) original ZERO

	LOCATION
CALL ZERO (DELOSX(1), DELOSZ(1100)	NUED
CALL ZERO(LENGTH(1),LENGTH(199))	NUED
CALL ZERO(XS(1),DZ(49))	BODYIN
CALL ZERO(BAREF,NYIG)	BODYIN
IF(BTHI.NE.1) CALL ZERO(VY(1),VY(NVYP))	BODYIN
CALL ZERO(XX(1),XZ(NXX,NYY))	BODGEOM
CALL ZERO(PAREF,ROTG)	PANELIN
CALL ZERO(CHORD(1),CHORD(20))	PANELIN
CALL INDEF(XOFF,ZOFF(200))	READIN
CALL INDEF(VX(1),JYIG(50))	BODYIN
CALL INDEF(PXL(1),VS(41))	PANELIN
CALL INDEF(A(LVX),A(LXX-1))	VORBOD
CALL INDEF(,A)	VORPANS

(b) original CALL's

Figure 2. Original and modified subroutine ZERO and CALL's

```
      SUBROUTINE ZERO(A,N)
      DIMENSION A(1)
      DATA PINDEF,HEXZRO/Z7F000000,Z00000000/
      STUFF=HEXZRO
10    DO 20 I=1,N
20    A(I)=STUFF
      RETURN
      ENTRY INDEF(A,N)
      STUFF=PINDEF
      GO TO 10
      END
```

(c) modified ZERO

```
                                                    LOCATION

      CALL ZERO(DELOSX(1),3300)                     NUED
      CALL ZERO(LENGTH(1,1),30)                     NUED
      CALL ZERO(XS(1),2008)                         BODYIN
      CALL ZERO(BAREF,29)                           BODYIN
      IF(BTHI.NE.1)CALL ZERO(VY(1),NVYP)            BODYIN
      CALL ZERO(XX(1),NXX+2*NXX*NYY)                BODGEOM
      CALL ZERO(PAREF,66)                           PANELIN
      CALL ZERO(CHORD(1),20)                        PANELIN
      CALL INDEF(XOFF,600)                          READIN
      CALL INDEF(VX(1),702)                         BODYIN
      CALL INDEF(PXL(1),2202)                       PANELIN
      CALL INDEF(A(LVX),LXX-LVX)                    VORBOD
      CALL INDEF(A,1)                               VORPANS
```

(d) modified CALL's

Figure 2. Concluded

```
      COMMON/RANF/ INDEX(200),LENGTH(199),
     X             GENDAT,VORTEX,CONPTS,NORMAL,
     X             CONFUN,VORTIN,PLANFM,TWICAM,
     X             THICKS,CPOINT,ELEVEN,TWELVE



      CALL OPENMS(19,INDEX,200,0)
```

(a) COMMON and initialization statements in NUED

```
      SUBROUTINE READRS(A,N,K,IC)
      COMMON/RANF/ INDEX(200),LENGTH(199),
     X             GENDAT,VORTEX,CONPTS,NORMAL,
     X             CONFUN,VORTIN,PLANFM,TWICAM,
     X             THICKS,CPOINT,ELEVEN,TWELVE
      EQUIVALENCE (CONPNT,CONPTS)
      DIMENSION A(N)
      DATA LUN/19/
      CALL=4HREAD
10    KEY=K
      IF(KEY.LE.12) KEY=12*(IC-1)+K
      LEN=N
      IF(LEN.EQ.0) LEN=LENGTH(KEY)
      IF(LEN.EQ.0) RETURN
      IF(LENGTH(KEY).EQ.0) LENGTH(KEY)=LEN
      IF(CALL.EQ.4HREAD) CALL READMS(LUN,A,LEN,KEY)
      IF(CALL.EQ.4HWRIT) CALL WRITMS(LUN,A,LEN,KEY,IREWR)
      RETURN
      ENTRY WRITRS
      CALL=4HWRIT
      IREWR=0
      IF(N.EQ.0) IREWR=-1
      GO TO 10
      END
```

(b) original READRS

Figure 3. Original and modified READRS and CALL's

```
                                                                    LOCATION
      CALL WRITRS(XOFF,SQUEEZE(XOFF,600,1),ELEVEN,1)                READIN
      CALL WRITRS(A,LEN,PLANFM,IBOD)                                BODYIN
      CALL WRITRS(JXIG,SQUEEZE(JXIG,350,1),CPOINT,IBOD)             BODYIN
      CALL WRITRS(VX,SQUEEZE(VX,352,1),VORTIN,IBOD)                 BODYIN
      CALL WRITRS(BAREF,29,GENDAT,IBOD)                             BODYIN
      CALL WRITRS(PXL,140,PLANFM,NBOD+IPAN)                         PANELIN
      CALL WRITRS(PXC,SQUEEZE(PXC,1020,1),TWICAM,NBOD+IPAN)         PANELIN
      CALL WRITRS(XOCT,SQUEEZE(XOCT,960,1),THICKS,NBOD+IPAN)        PANELIN
      CALL WRITRS(VC,82,VORTIN,NBOD+IPAN)                           PANELIN
      CALL WRITRS(PAREF,66,GENDAT,NBOD+IPAN)                        PANELIN
      CALL READRS(BAREF,0,GENDAT,I)                                 VORBOD
      CALL READRS(A(LXX),0,PLANFM,I)                                VORBOD
      CALL READRS(A,0,VORTIN,I)                                     VORBOD
      CALL READRS(A(LXIG),0,CPOINT,I)                               VORBOD
      CALL READRS(PAREF,0,GENDAT,I)                                 VORPANS
      CALL READRS(PXL,0,PLANFM,I)                                   VORPANS
      CALL READRS(VC,0,VORTIN,I)                                    VORPANS
      CALL READRS(Z(KPX),0,TWICAM,I)                                VORPANS
      CALL READRS(A(KXT),0,THICKS,I)                                VORPANS
      CALL READRS(A(KXOFF),0,ELEVEN,1)                              VORPANS
      CALL READRS(PAREF,0,GENDAT,II)                                BPCOEF
```

(c) original CALL's

```
      COMMON/RANF/ INDEX(6,5),LENGTH(6,5),STORE(317,5),
     X             LOCINT(4),LOCFIN(4),IDACSS
```

```
      DEFINE FILE 19(384,100,U,IDACSS)
      IDACSS=1
      LOCINT(1)=1
      LOCINT(2)=30
      LOCINT(3)=170
      LOCINT(4)=252
      LOCFIN(1)=29
      LOCFIN(2)=169
      LOCFIN(3)=251
      LOCFIN(4)=317
```

(d) modified COMMON and initialization statements in NUED

Figure 3. Continued

```
      SUBROUTINE READRS(A,N,K,IC)
      COMMON/RANF/ INDEX(6,5),LENGTH(6,5),STORE(317,5),
     X             LOCINT(4),LOCFIN(4),IDACSS
      DIMENSION A(N)
      DATA LUN/19/,READ,WRIT/4HREAD,4HWRIT/
      CALL=READ
10    IF(K.GT.6) GO TO 50
      LEN=N
      IF(LEN.EQ.0) LEN=LENGTH(K,IC)
      IF(LEN.EQ.0) RETURN
      IF(LENGTH(K,IC).EQ.0) LENGTH(K,IC)=LEN
      IF(CALL.EQ.READ) GO TO 25
      IF(IDACSS.NE.1) IDACSS=NXTWRT
      INDEX(K,IC)=IDACSS
      DO 15 I=1,LEN,100
      CALL WRITMS(LUN,A(I),IDACSS)
15    CONTINUE
      NXTWRT=IDACSS
      RETURN
25    IDACSS=INDEX(K,IC)
      DO 30 I=1,LEN,100
      CALL READMS(LUN,A(I),IDACSS)
30    CONTINUE
      RETURN
50    II=LOCINT(K-6)
      IF=LOCFIN(K-6)
      IF(CALL.EQ.READ) GO TO 65
      DO 60 I=II,IF
      STORE(I,IC)=A(I-II+1)
60    CONTINUE
      RETURN
65    DO 70 I=II,IF
      A(I-II+1)=STORE(I,IC)
70    CONTINUE
      RETURN
      ENTRY WRITRS(A,N,K,IC)
      CALL=WRIT
      GO TO 10
      END
```

(e) modified READRS

Figure 3. Continued

```
      SUBROUTINE RANDIO
      DIMENSION A(100)
      ENTRY READMS(LUN,A,/IDACSS/)
      READ(LUN'IDACSS)A
      RETURN
      ENTRY WRITMS(LUN,A,/IDACSS/)
      WRITE(LUN'IDACSS)A
      RETURN
      END
```

(f) replacement subroutine for CDC READMS and WRITMS functions

	LOCATION
CALL WRITRS(XOFF,SQUEEZ(XOFF,600,1),1,1)	READIN
CALL WRITRS(A,LEN,2,IBOD)	BODYIN
CALL WRITRS(JXIG,SQUEEZ(JXIG,350,1),3,IBOD)	BODYIN
CALL WRITRS(VX,SQUEEZ(VX,352,1),4,IBOD)	BODYIN
CALL WRITRS(BAREF,29,7,IBOD)	BODYIN
CALL WRITRS(PXL,140,8,IPAN)	PANELIN
CALL WRITRS(PXC,SQUEEZ(PXC,1020,1),5,IPAN)	PANELIN
CALL WRITRS(XOCT,SQUEEZ(XOCT,960,1),6,IPAN)	PANELIN
CALL WRITRS(VC,82,9,IPAN)	PANELIN
CALL WRITRS(PAREF,66,10,IPAN)	PANELIN
CALL READRS(BAREF,0,7,I)	VORBOD
CALL READRS(A(LXX),0,2,I)	VORBOD
CALL READRS(A,0,4,I)	VORBOD
CALL READRS(A,LXIG),0,3,I)	VORBOD
CALL READRS(PAREF,0,10,II)	VORPANS
CALL READRS(PXL,0,8,II)	VORPANS
CALL READRS(VC,0,9,II)	VORPANS
CALL READRS(A(KPX),0,5,II)	VORPANS
CALL READRS(A(KXT),0,6,II)	VORPANS
CALL READRS(A(KXOFF),0,1,1)	VORPANS
CALL READRS(PAREF,0,10,I)	BPCOEF

(g) modified CALL's

Figure 3. Concluded

```
      IF ( NSYM - 1 ) 58,52,54
52    ASSIGN 910 TO I19
      GO TO 60
54    ASSIGN 920 TO I19
      GO TO 60
58    ASSIGN 2000 TO I19
```

```
      DO 1700 I2 = 1, LOOP
      IF ( I2 .EQ.LOOP ) GO TO I19, ( 2000, 910, 920)
      GO TO (1000, 910, 920, 910 ), I2
```

(a) original

```
      DO 1700 I2 = 1, LOOP
      IF(I2.EQ.LOOP) GO TO 880
      IF(I2-2)1000,910,870
870   IF(I2-4)920,910,880
880   IF(NSYM-1)2000,910,920
```

(b) modified

Figure 4. Original and modified coding in subroutines VFMNLF and VFMLFT

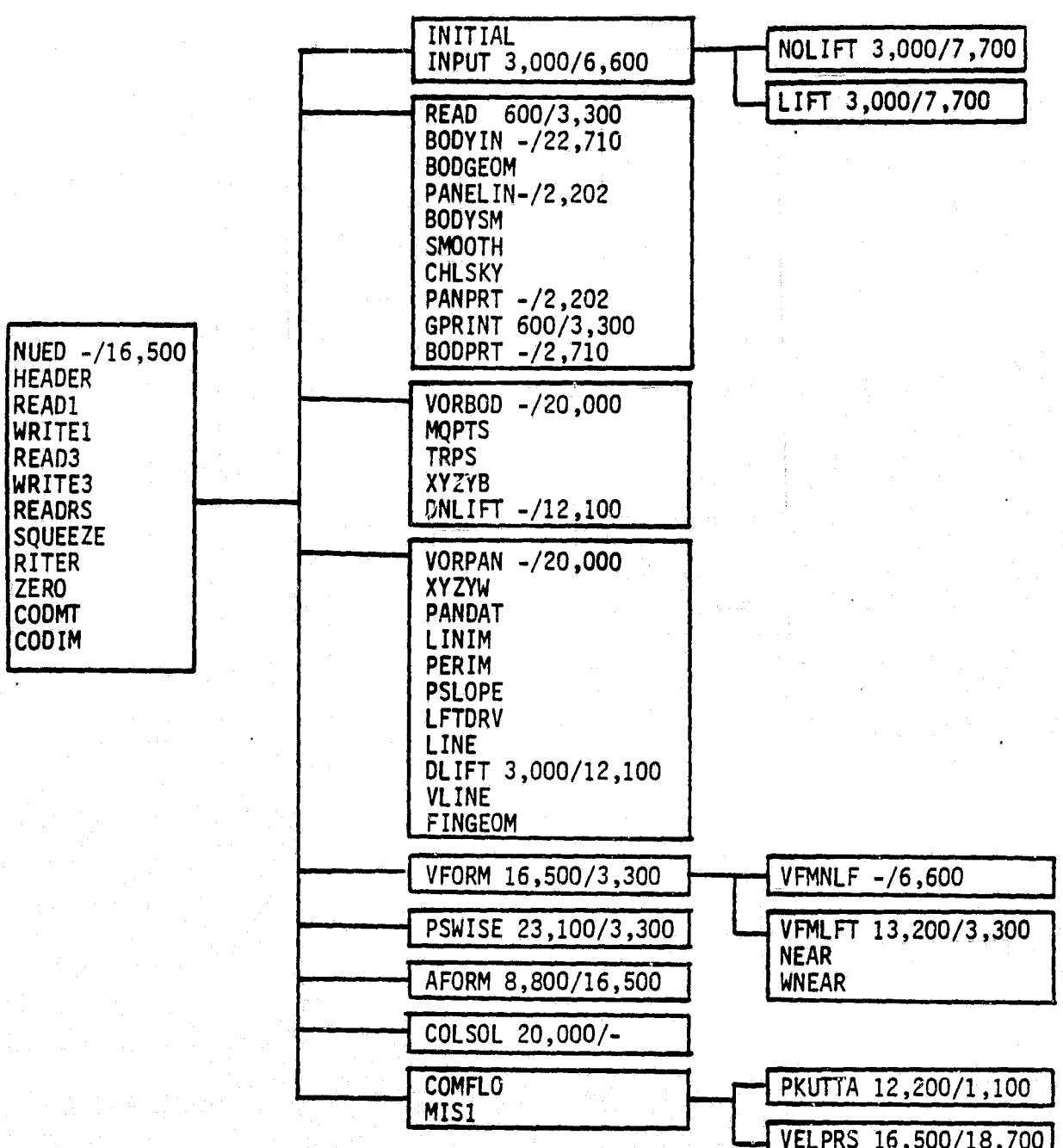


Figure 5. Overlay structure. Numbers denote size of DIMENSION and COMMON respectively

```
      COMMON / VARONSF / DELOSX(1100), DELOSY(1100), DELOSZ(1100)
      COMMON / CTABLE / XC(1100), YC(1100), ZC(1100)
      COMMON / NORMAL / XN(1100), YN(1100), ZN(1100)
      COMMON / ATABLE / A(1100)
      COMMON / XYZAGT/ XNGT(1100), YNGT(1100), ZNGT(1100),AGT(1100)
      COMMON / STABLE / SIGMA(1100)
```

(a) original COMMON blocks

```
      COMMON / BLANK/DELOSX(1100),DELOSY(1100),DELOSZ(1100),
     X               XC(1100),YC(1100),ZC(1100),
     X               XN(1100),YN(1100),ZN(1100),
     X               A(1100),
     X               XNGT(1100),YNGT(1100),ZNGT(1100),AGT(1100),
     X               SIGMA(1100),
     X               DUMMY(26400)
```

(b) modified COMMON block

```
      COMMON /BLANK/DELOSX(1100),DELOSY(1100),DELOSZ(1100),
     X               DUMMY(13200),
     X               XOFF(200),YOFF(200),ZOFF(200),
     X               DUMMY2(25800)
```

(c) use in READIN and GPRINT

```
      COMMON /BLANK/DUMMY1(16500),
     X         XS(41),TYS(21,41),RZS(21,41),XMC(49),YM(49),ZM(49),
     X         DY(49),DZ(49),VX(301),VY(51),
     X          JXIG(300),JYIG(50),DUMMY2(100),A(23590)
```

(d) use in BODYIN and BODPRT

Figure 6. Modification to large array storage requirements

```
      COMMON /BLANK/DUMMY1(16500),
     X       PXL(20),PYL(20),PZL(20),CHORD(20),PXT(20),PYT(20),
     X       PZT(20),PXC(30),PET(30),CAMBER(30,30),TWIST(30),
     X       TWPCR(30),XOCT(30),ETAT(30),THICK(30,30),
     X       VC(41),VS(41),
     X       DUMMY2(24198)
```

(e) use in PANELIN and PANPRT

```
      COMMON /BLANK/DUMMY(16500),A(26400)
```

(f) use in VORBOD and VORPAN

```
      COMMON / BLANK / DELOSX(1100), DELOSY(1100), DELOSZ(1100),
     X                 DUMMY1(13200),
     X            VXS    (1100 ) ,     VYS    (1100 ) ,
     X            VZS    (1100 ) ,     DIPLEX (1100 ) ,
     X            DIPLEY (1100 ) ,     DIPLEZ (1100 ) ,
     X            VXL    (1100 ) ,     VYL    (1100 ) ,
     X            VZL    (1100 ) ,     VXR    (1100 ) ,
     X            VYR    (1100 ) ,     VZR    (1100 ) ,
     X            VI     (1100 ) ,     VJ     (1100 ) ,
     X            VK     (1100 ) ,     DUMMY2 (9900 )
```

(g) use in VFORM

```
      COMMON /BLANK/DUMMY1(3300),
     X              XC(1100),YC(1100),ZC(1100),
     X              DUMMY2(9900),
     X              XIJ(1100),YIJ(1100),ZIJ(1100),
     X              DUMMY3(23100)
```

(h) use in VFMNLF

Figure 6. Continued

```
      COMMON /BLANK/DUMMY1(3300),
     X              XC(1100),YC(1100),ZC(1100),
     X              DUMMY2(9900),
     X              XIJ    (1100 ) ,    YIJ    (1100 ) ,
     X              ZIJ    (1100 ) ,    VXL    (1100 ) ,
     X              VYL    (1100 ) ,    VZL    (1100 ) ,
     X              VXR    (1100 ) ,    VYR    (1100 ) ,
     X              VZR    (1100 ) ,    VXTRAX (1100 ) ,
     X              VXTRAY (1100 ) ,    VXTRAZ (1100 ) ,
     X              DUMMY3(13200)
```

(i) use in VFMLFT

```
      COMMON  / BLANK / DELOSX(1100), DELOSY(1100), DELOSZ(1100),
                        DUMMY1(13200),
     X              VSUMX  (1100 ) ,    VSUMY  (1100 ) ,
     X              VSUMZ  (1100 ) ,    VDIFX  (1100 ) ,
     X              VDIFY  (1100 ) ,    VDIFZ  (1100 ) ,
     X              VLASTX (1100 ) ,    VLASTY (1100 ) ,
     X              VLASTZ (1100 ) ,    VNEXTX (1100 ) ,
     X              VNEXTY (1100 ) ,    VNEXTZ (1100 ) ,
     X              ONSETX (1100 ) ,    ONSETY (1100 ) ,
     X              ONSETZ (1100 ) ,
     X              VXL    ( 1100) ,    VYL    (1100 ) ,
     X              VZL    ( 1100) ,    VXR    (1100 ) ,
     X              VYR    ( 1100) ,    VZR    (1100 ) ,
     X              DUMMY2(3300)
```

(j) use in PSWISE

```
      COMMON /BLANK/DELOSX(1100),DELOSY(1100),DELOSZ(1100),
     X              XC(1100),YC(1100),ZC(1100),
     X              XN(1100),YN(1100),ZN(1100),
     X              A(1100),
     X              XNGT(1100),YNGT(1100),ZNGT(1100),AGT(1100),
     X              SIGMA(1100),
     X              VX     (1100 ) ,    VY     (1100 ) ,
     X              VZ     (1100 ) ,    RHSIDE (1100 ) ,
     X             AIJ     (1100 ) ,
     X              DIPLEX (1100) ,     DIPLEY (1100) ,
     X              DIPLEZ (1100) ,
     X              DUMMY(17600)
```

(k) use in AFORM

Figure 6. Continued

```
      COMMON /BLANK/A(42900)
```

(l) use in COLSOL

```
      COMMON /BLANK/DUMMY1(15400),SIGMA(1100),
     X                FLOWX  (  40  )    ,     FLOWY   ( 40 )  ,
     X                FLOWZ  (  40  )    ,     VELX    ( 40 )  ,
     X                VELY   (  40  )    ,     VELZ    ( 40 )  ,
     X                SUMX   (  40  )    ,     SUMY    ( 40 )  ,
     X                SUMZ   (  40  )    ,     VXINF   ( 40 )  ,
     X                VYINF  (  40  )    ,     VZINF   ( 40 )  ,
     X                VXFLOW ( 40, 20 )  ,     VYFLOW  ( 40, 20 ) ,
     X                VZFLOW ( 40, 20 )  ,     DVSQ    ( 20 )  ,
     X                TERM1  (  20  )    ,     S       ( 20, 20 ) ,
     X                PS     (  20  )    ,     P       (
     X                DELB     (   20  )  .       D       ( 20. 20 ) .
     X                EMINF  ( 2 )  .
     X                EM     (  2. 20 )  .     QINF    ( 20, 20 )  ,
     X                Q(20,20,20)
     X                DUMMY2(14178)
```

(m) use in PKUTTA

```
      COMMON /BLANK/DUMMY1(6600),
     X                XN(1100),YN(1100),ZN(1100),
     X                DUMMY2(1100),
     X                XNGT(1100),YNGT(1100),ZNGT(1100),AGT(1100),
     X                SIGMA(1100),
     X                   COMSIG(1100), VX(1100), VY(1100), VZ(1100),
     X                    VEL(1100), DCX(1100), DCY(1100), DCZ(1100),
     X                    VN(1100), PCOEF(1100), VSUM(1100),
     X                ONSETX(1100),ONSETY(1100),ONSETZ(1100),
     X                CFLOWX(1100),CFLOWY(1100),CFLOWZ(1100),
     X                VXS(1100),VYS(1100),VZS(1100),
     X                UNIFMX(1100),UNIFMY(1100),UNIFMZ(1100),
     X                USIGMA(1100)
```

(n) use VELPRS

Figure 6. Continued

```
      COMMON /BLANK/DUMMY1(3300),
     X                 XC(1100),YC(1100),ZC(1100),
     X                 XN(1100),YN(1100),ZN(1100),
     X                 A(1100),
     X                 DUMMY2(1100),
     X                 XNGT(1100),YNGT(1100),ZNGT(1100),AGT(1100),
     X                     COMSIG(1100), VX(1100), VY(1100), VZ(1100),
     X                     VEL(1100), DCX(1100), DCY(1100), DCZ(1100),
     X                     VN(1100), PCOEF(1100), VSUM(1100),
     X                      MXI(200), MYJ(200), MZK(200),
     X                      FSTRPX(200), FSTRPY(200), FSTRPZ(200),
     X                      SECMOX(50), SECMOY(50), SECMOZ(50),
     X                      FSECX(50), FSECY(50), FSECZ(50),
     X                      XMTOT, YMTOT, ZMTOT, XFTOT, YFTOT, ZFTOT,
     X                      DUMMY3(12794)
```

(o) use in PRINT

```
      COMMON /BLANK/DUMMY1(3300),
     X                 XC(1100), YC(1100), ZC(1100),
     X                 DUMMY2(22000),
     X                      MXI(200), MYJ(200), MZK(200),
     X                      FSTRPX(200), FSTRPY(200), FSTRPZ(200),
     X                      SECMOX(50), SECMOY(50), SECMOZ(50),
     X                      FSECX(50), FSECY(50), FSECZ(50),
     X                      XMTOT, YMTOT, ZMTOT, XFTOT, YFTOT, ZFTOT,
     X                      DUMMY3(12794)
```

(p) use in BPCOEF

Figure 6. Concluded

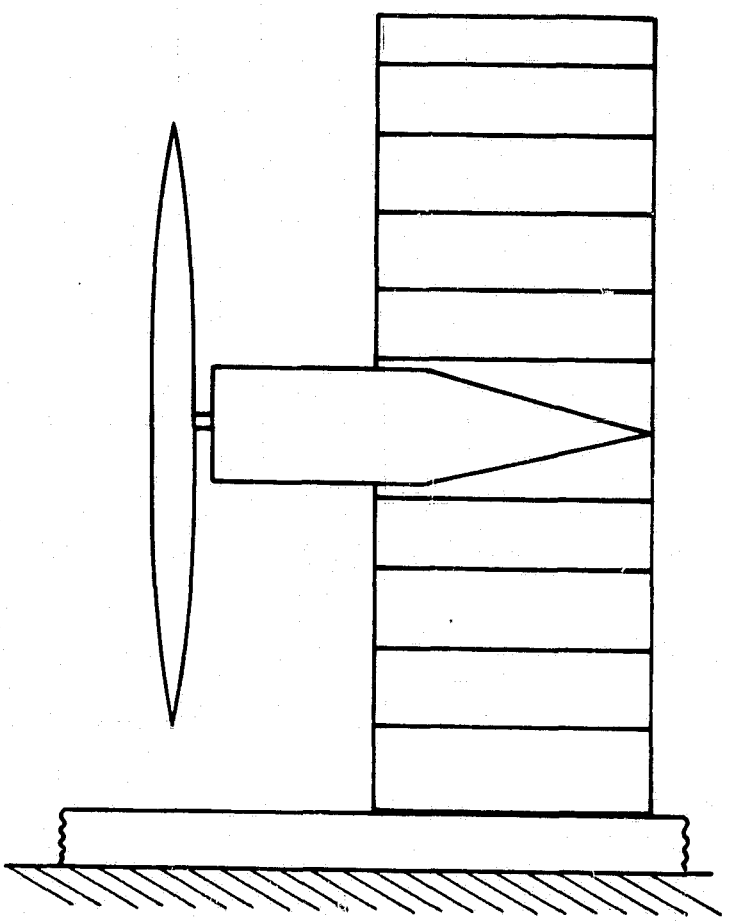
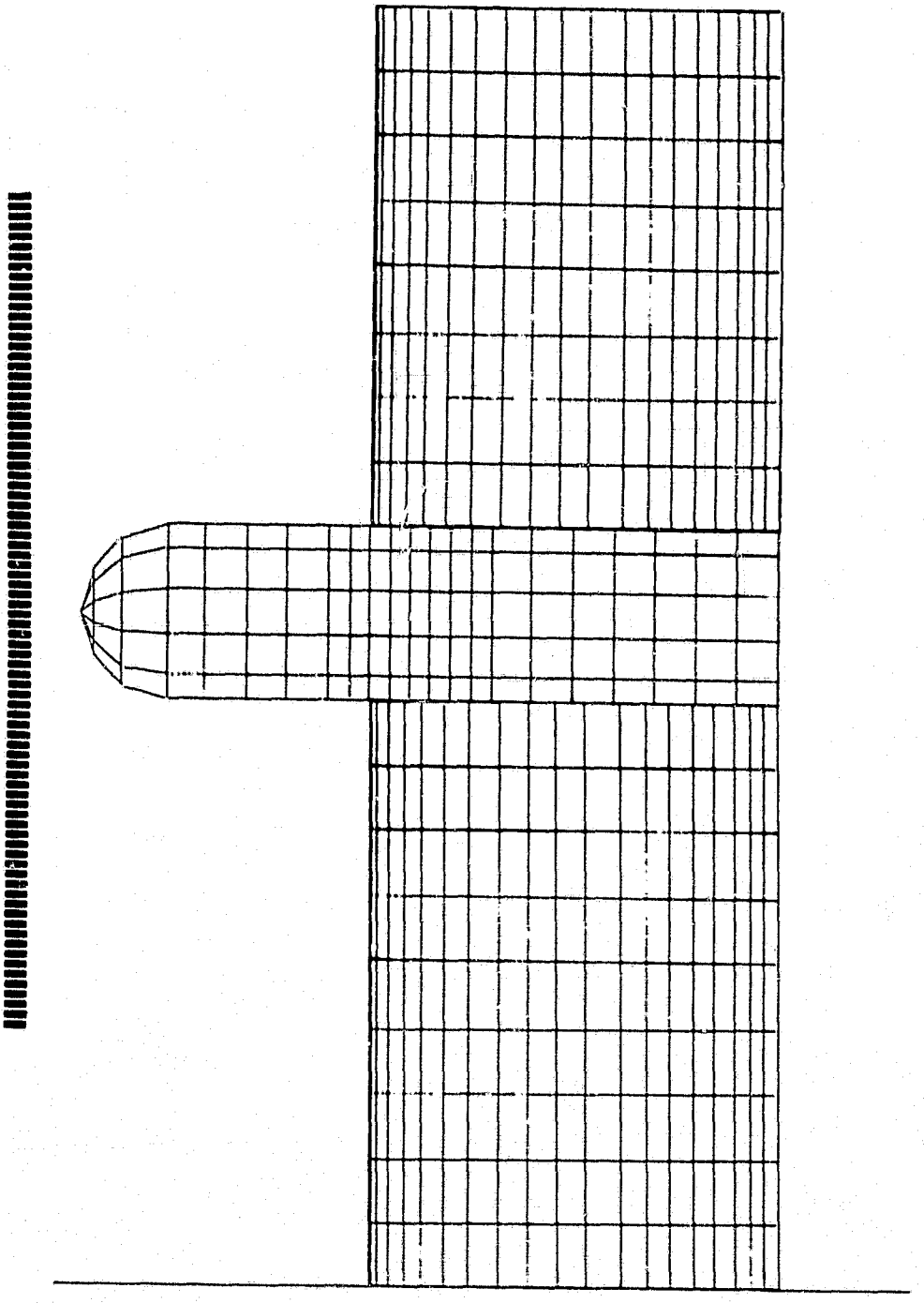
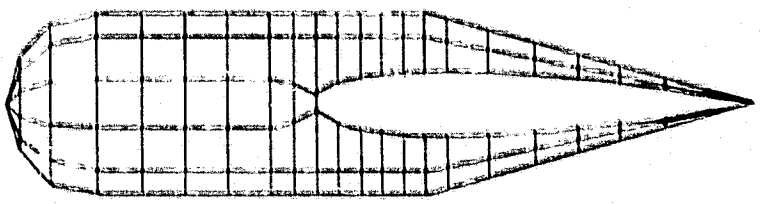


Figure 7. Reflection plane test model from Reference (3).



(a) Top view

Figure 8. Panel layout for the test model of Reference (3).



(b) Side view

(c) Airfoil

Figure 8. Concluded

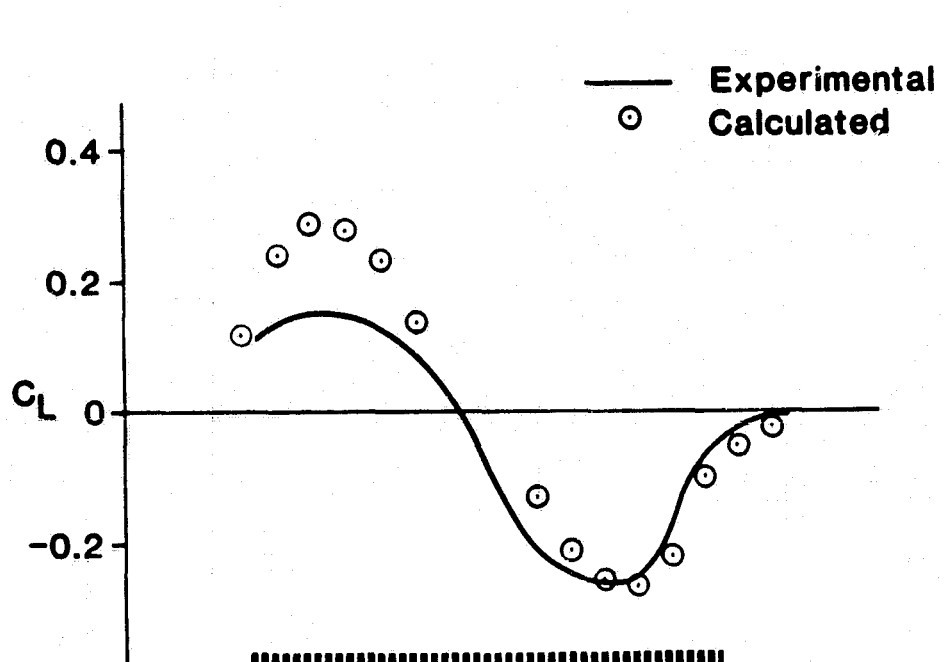


Figure 9. Comparison with Reference (3), calculated versus experimental.

(a) Reference Figure 24, $\alpha=0°$, $C_{T_S} = 0.6$

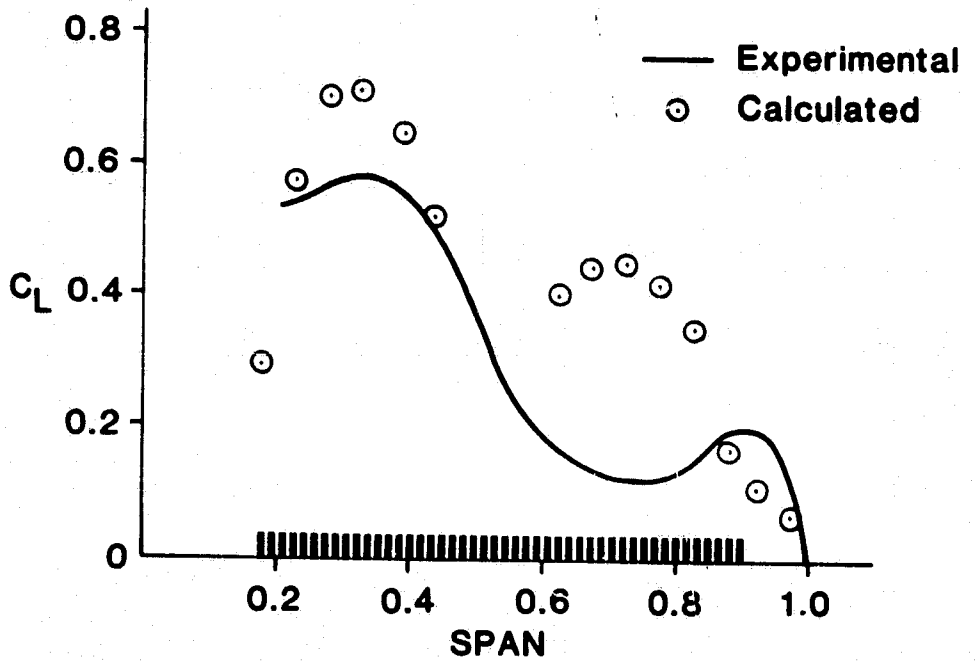


Figure 9. continued

(b) Reference Figure 24, $\alpha=10°$, $C_{T_S} = 0.6$

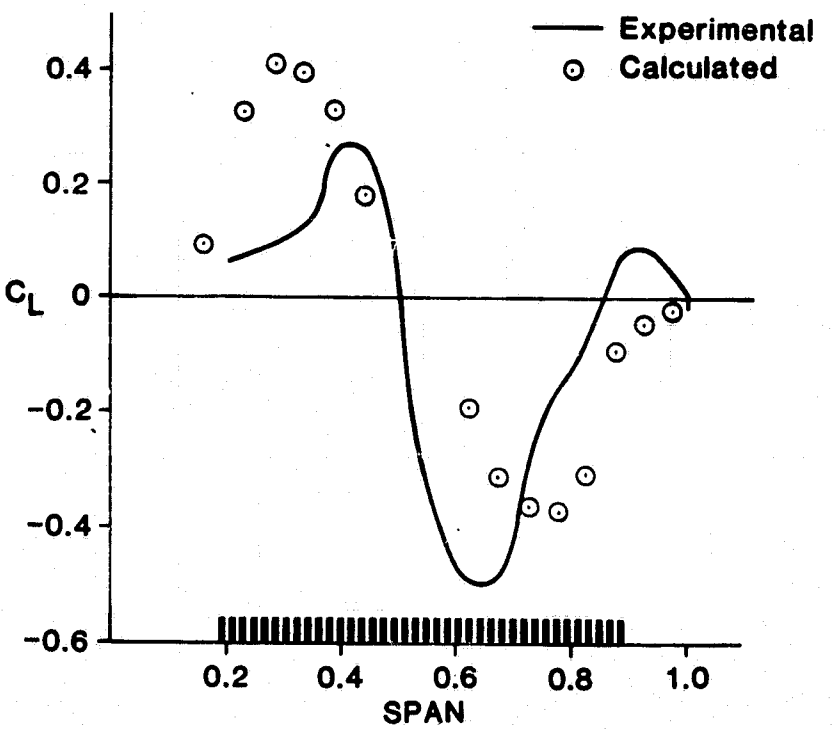


Figure 9. continued

(c) Reference Figure 42, $\alpha=0°$, $C_{T_S} = 0.5$

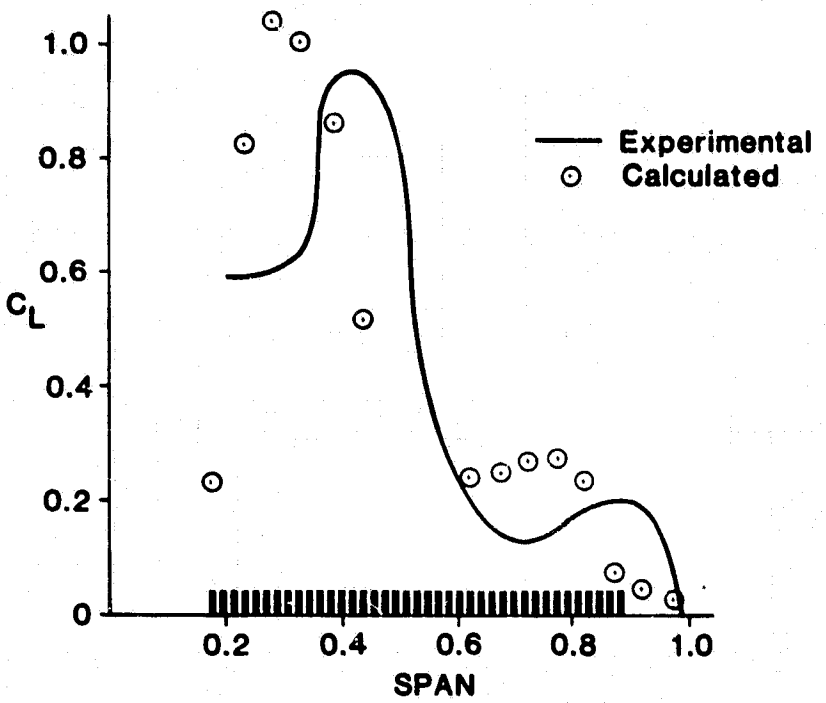


Figure 9. concluded

(d) Reference Figure 42 , α=10°, C_{T_S} = 0.5

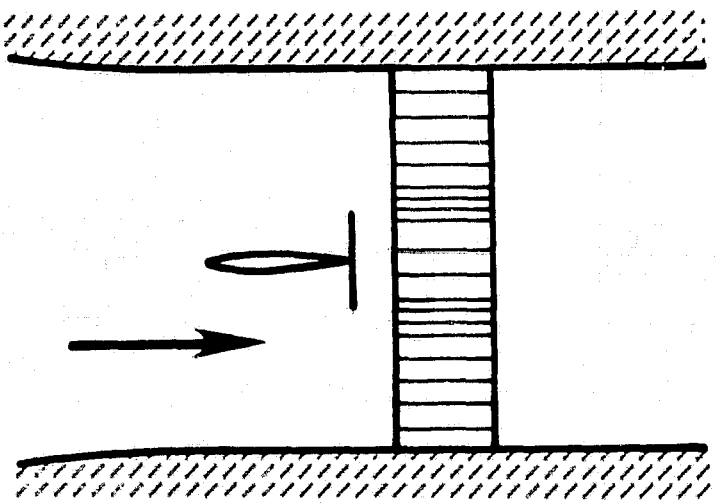
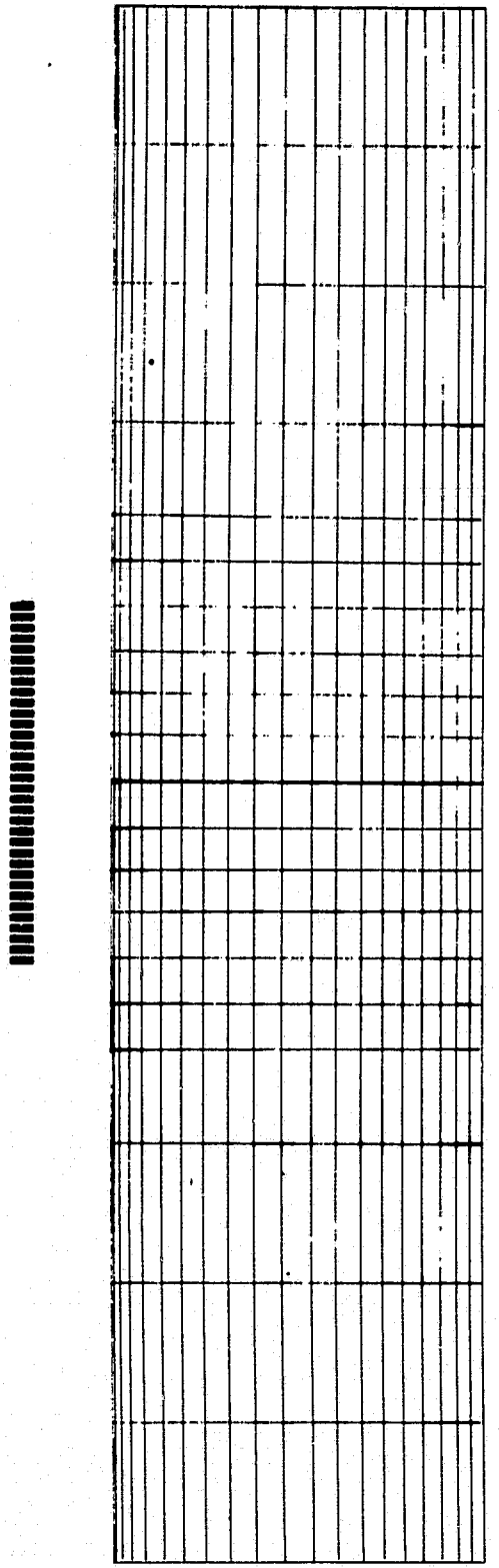
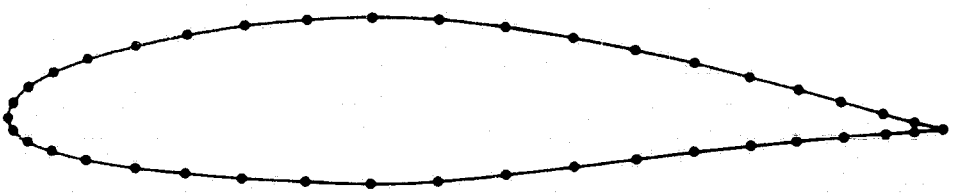


Figure 10. Infinite aspect ratio test model from Reference (5).



(a) Top view

Figure 11. Panel layout for the test model of Reference (5)



(b) Airfoil

Figure 11. Concluded

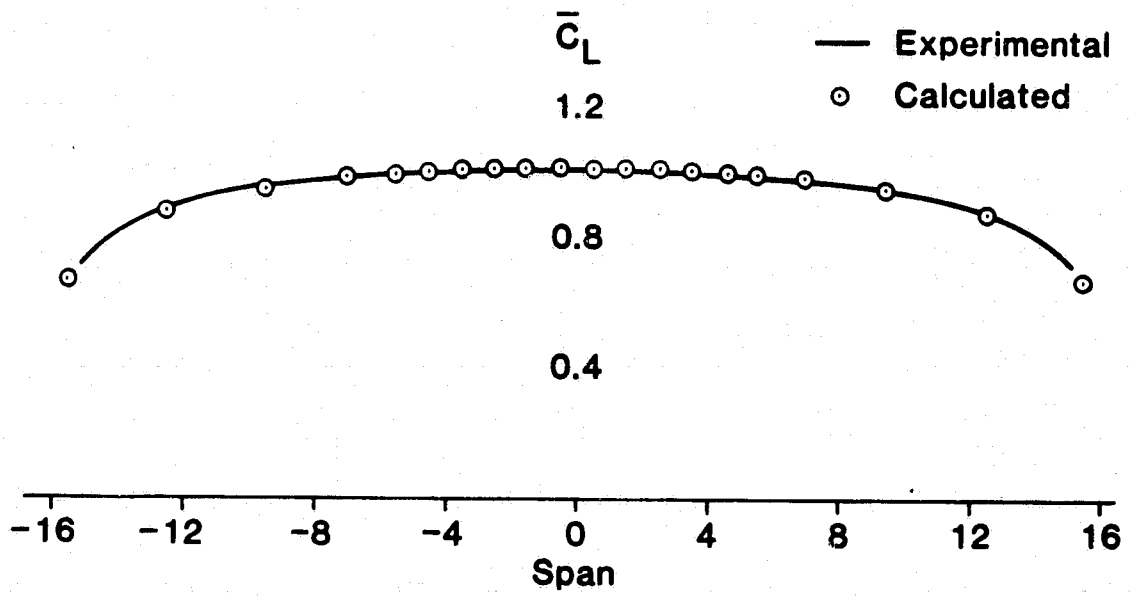


Figure 12. Comparison with Reference (5), calculated versus experimental.

(a) Reference Figure 10a, $\alpha=6\frac{1}{2}°$, $\sigma = 0.0$

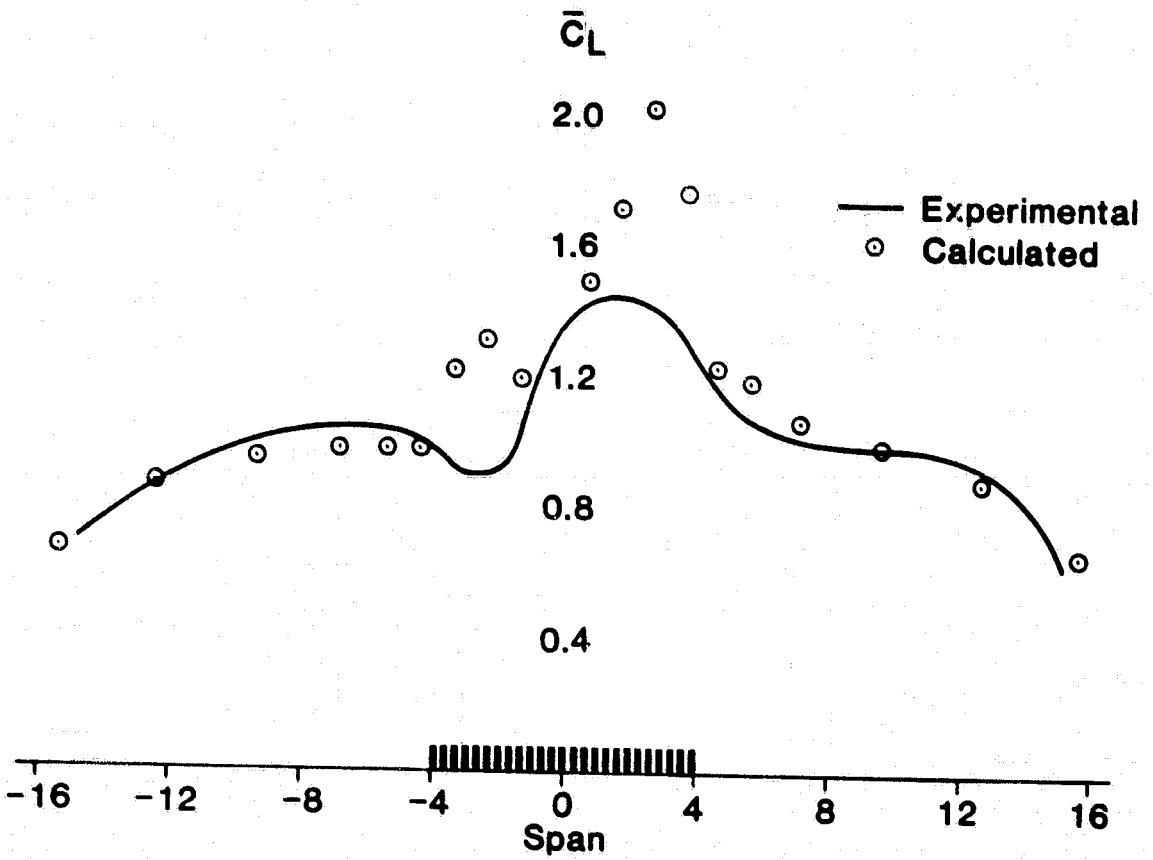


Figure 12. Continued

(b) Reference Figure 10a, $\alpha=6\frac{1}{2}°$, σ = 0.26

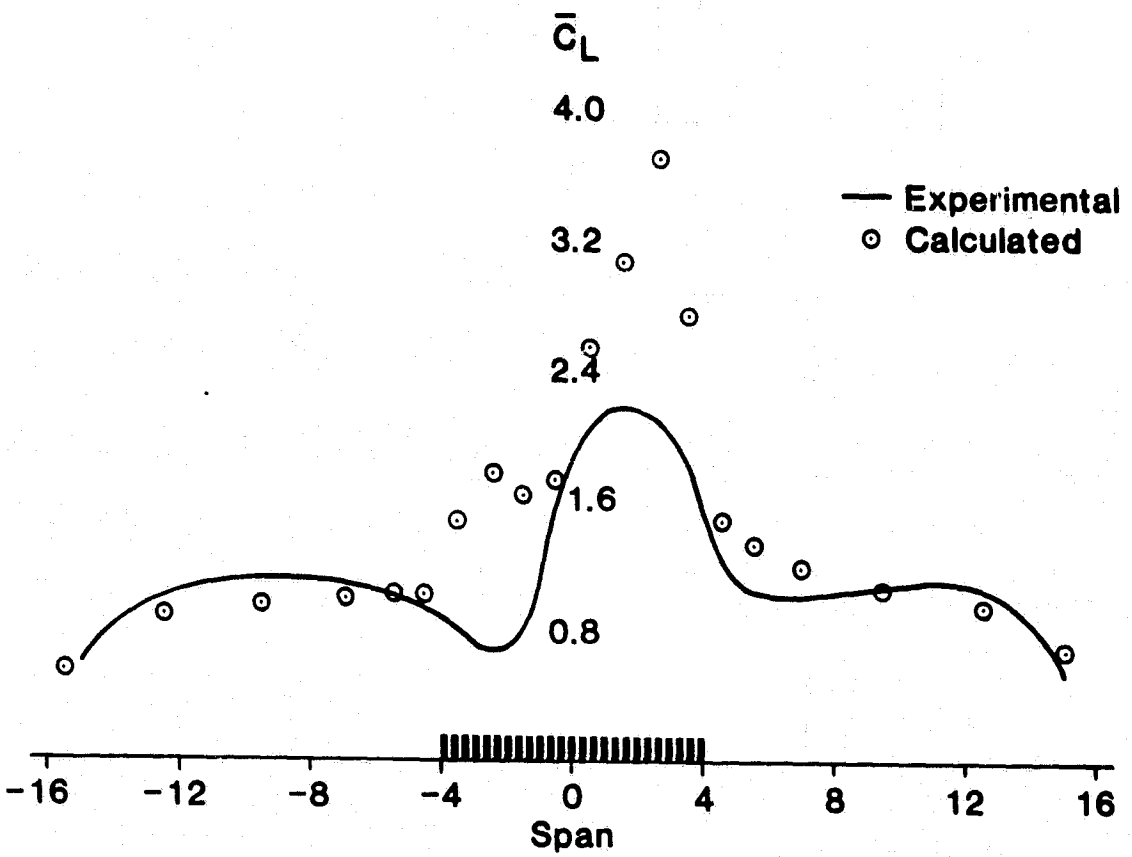


Figure 12. Concluded

(c) Reference Figure 10b, $\alpha=6\frac{1}{2}°$, σ = 0.49

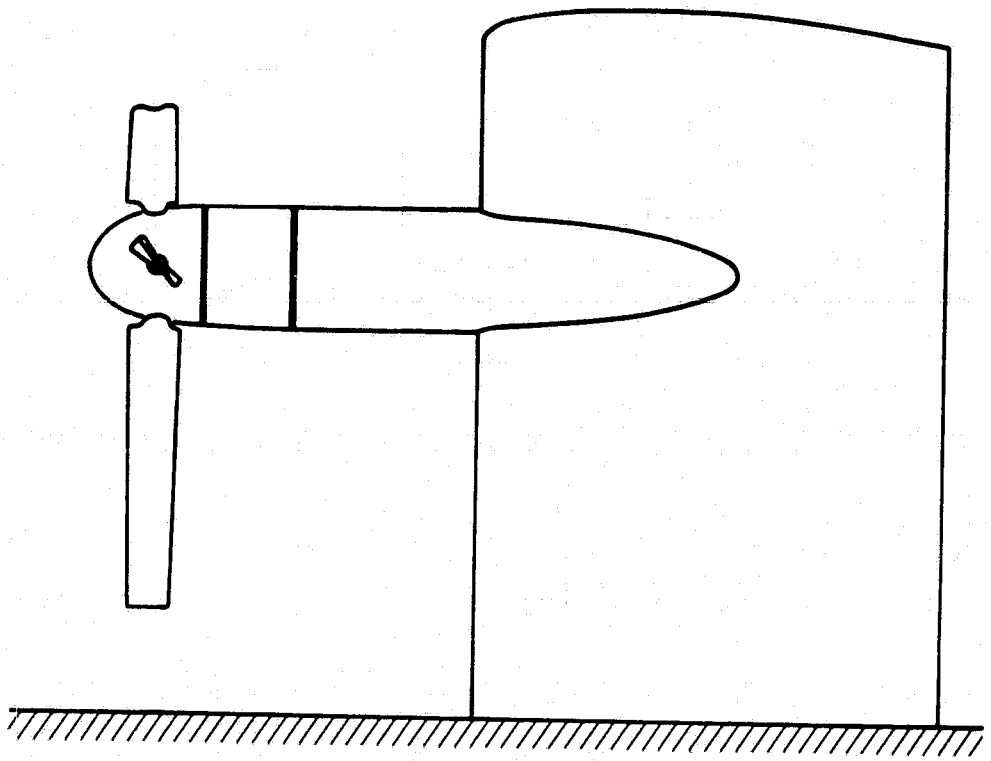
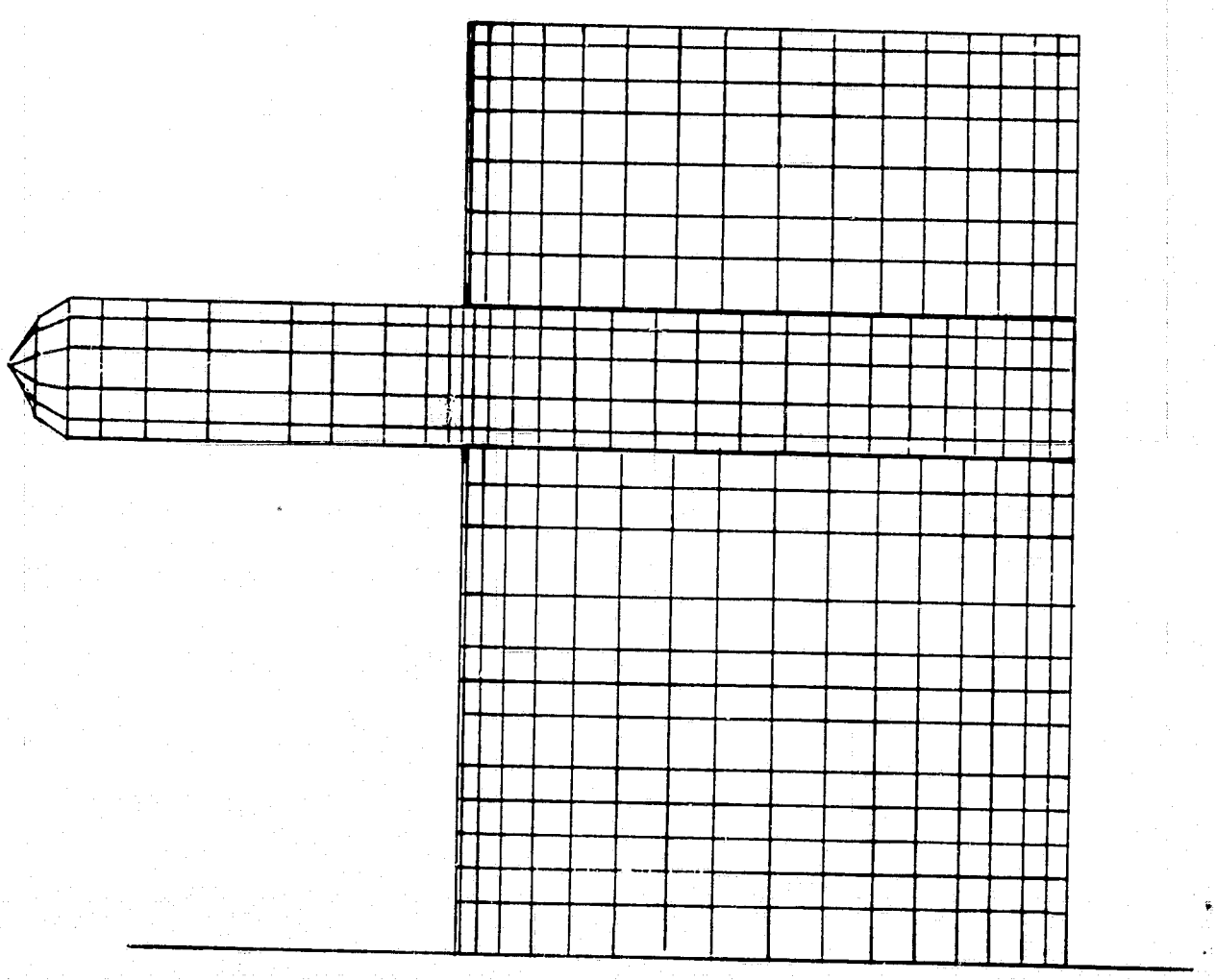
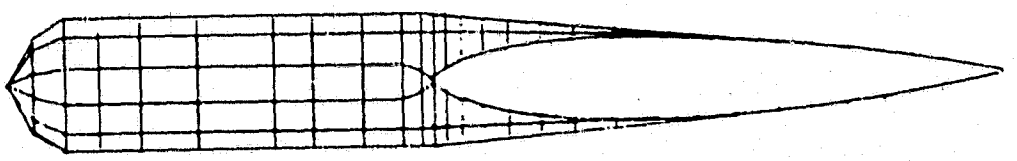


Figure 13. Reflection plane test model from Reference (6).

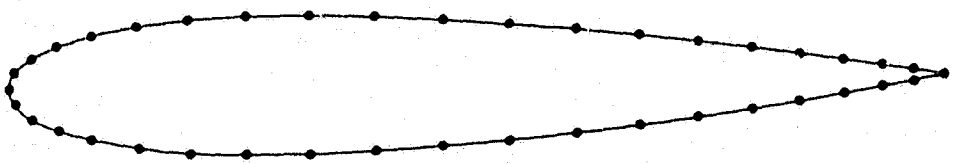


(a) Top view

Figure 14. Panel layout for the test model of Reference (6).



(b) Side view



(c) Airfoil

Figure 14. Concluded

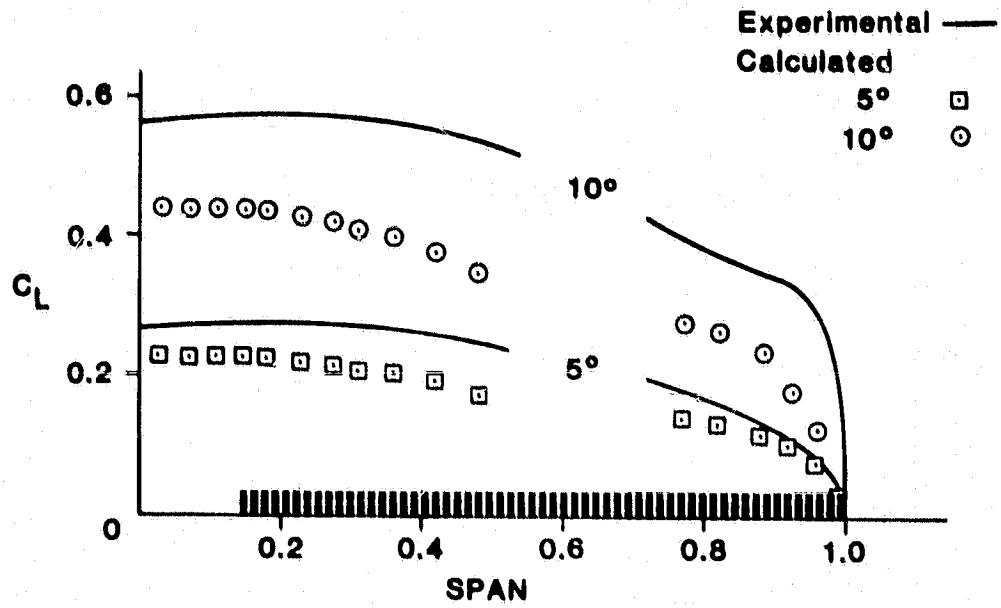


Figure 15. Comparison with Reference (6), calculated versus experimental.

(a) Reference tables 5 and 6, α=5°,10°, C_T=0

ORIGINAL PAGE IS
OF POOR QUALITY

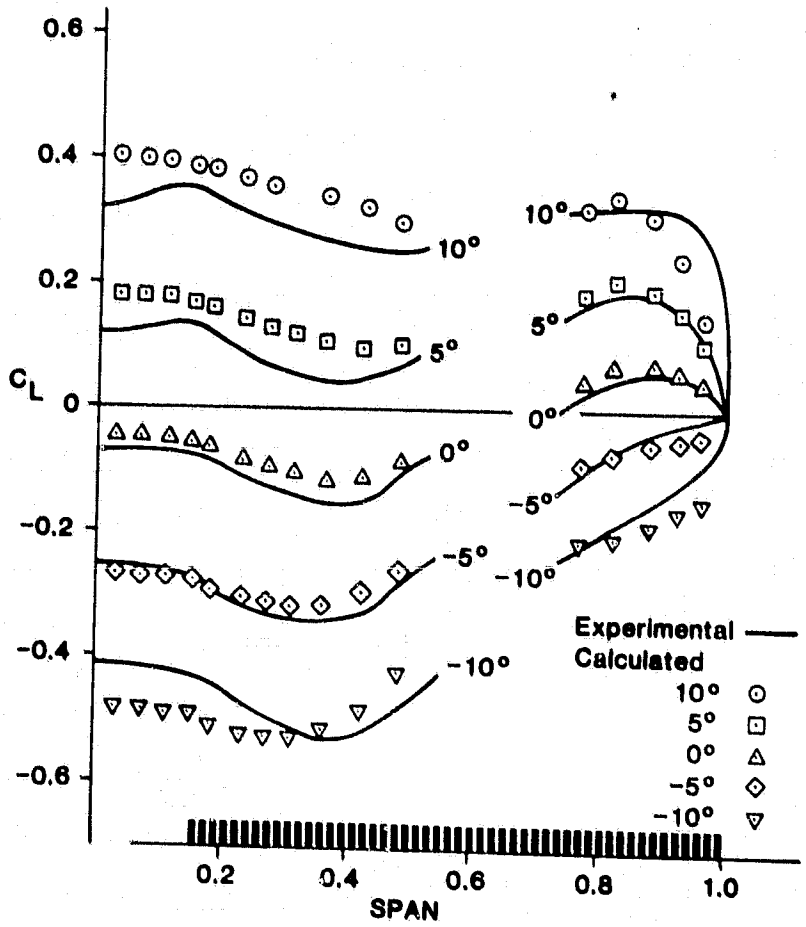


Figure 15. Continued

(b) Reference tables 7-11, α=-10°, -5°, 0°, 5°, 10°; C_T=0.36

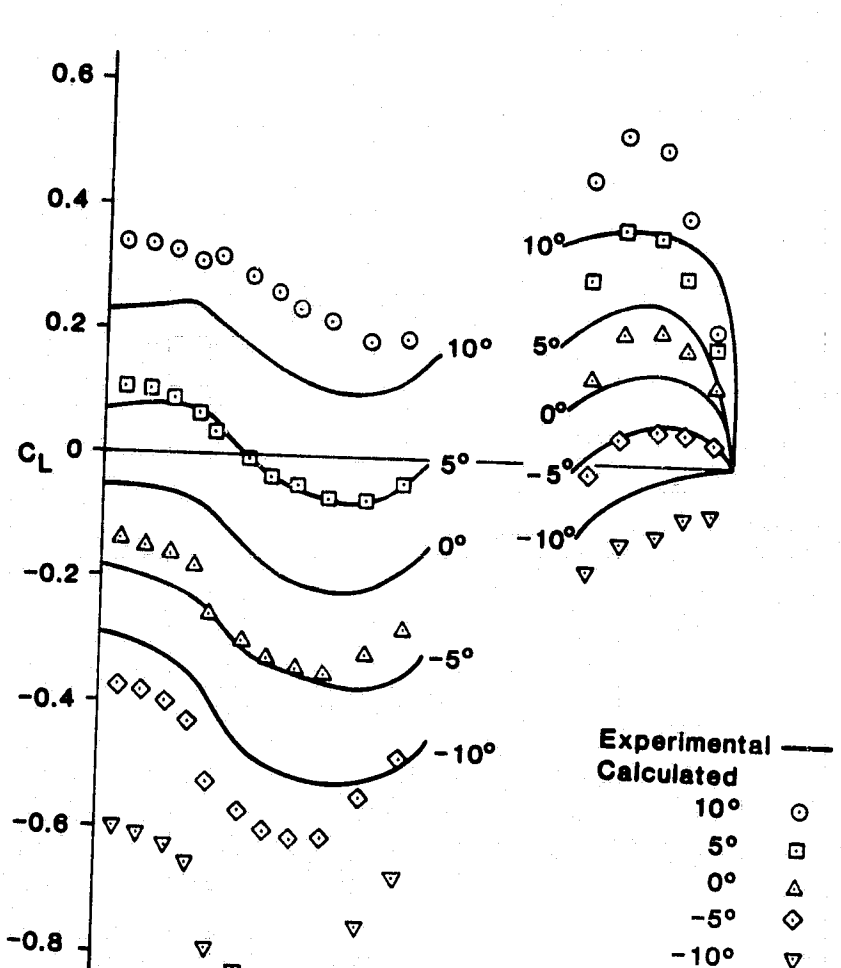


Figure 15. Concluded

(c) Reference table 12-16, α=-10°, -5°, 0°, 5°, 10°; C_T=0.64

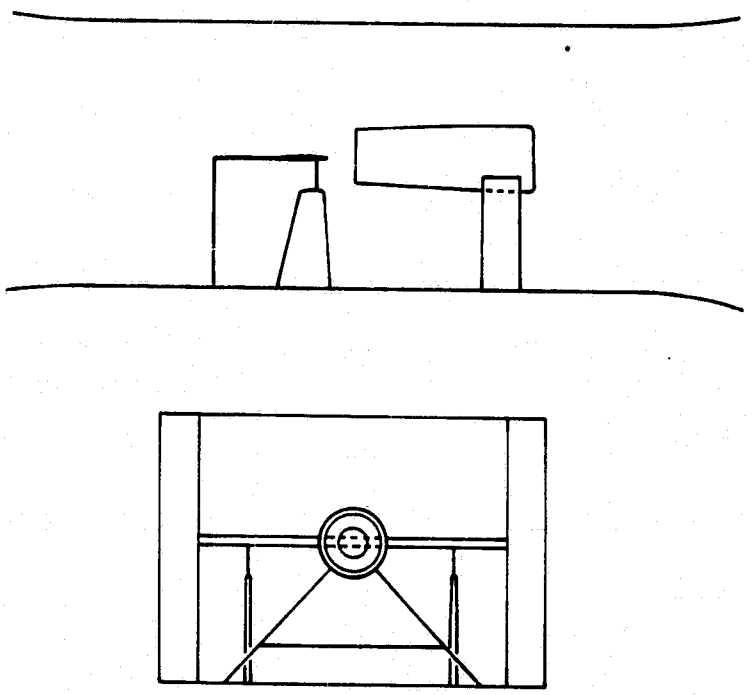
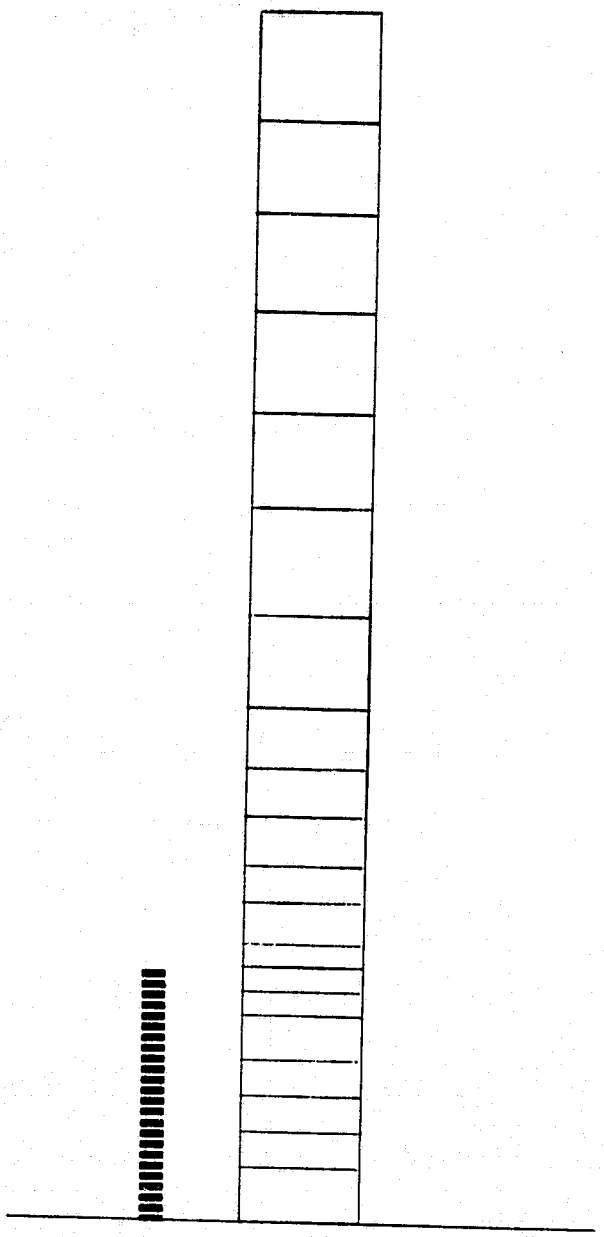
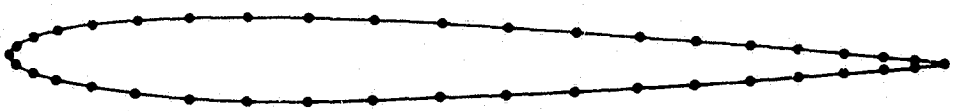


Figure 16. Infinite aspect ratio test model from Reference (7).



(a) Top view

Figure 17. Spanwise panel layout for the test model of Reference (7).



(b) Airfoil

Figure 17. Concluded

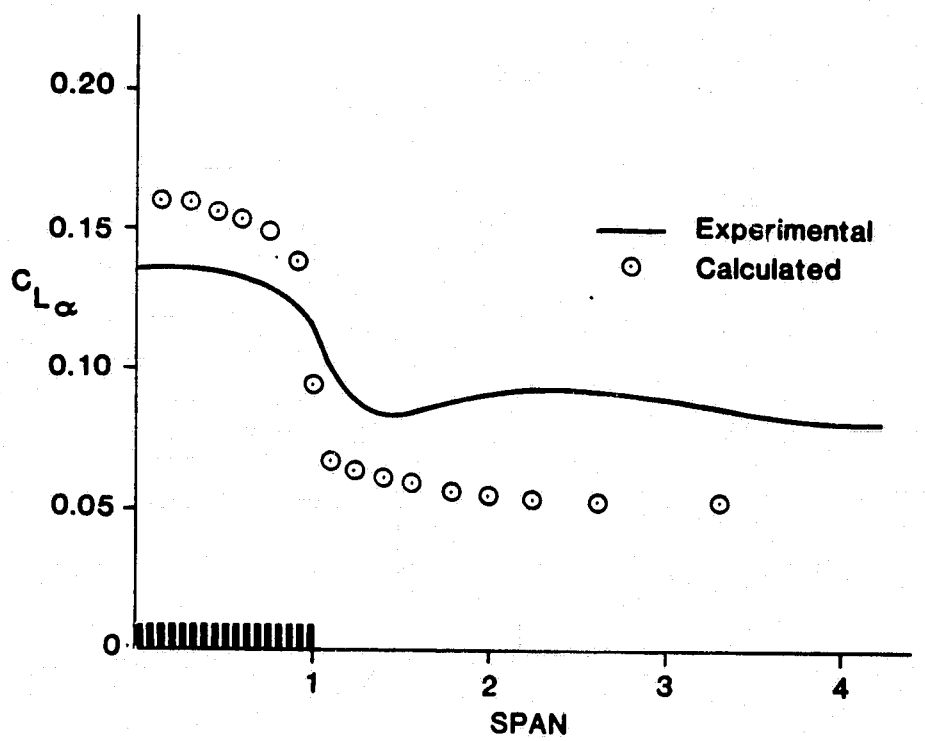


Figure 18. Comparison with Reference (7), calculated versus experimental.

(a) Reference Figure V-5d, $V_{jet}/V_\infty = 2.0$

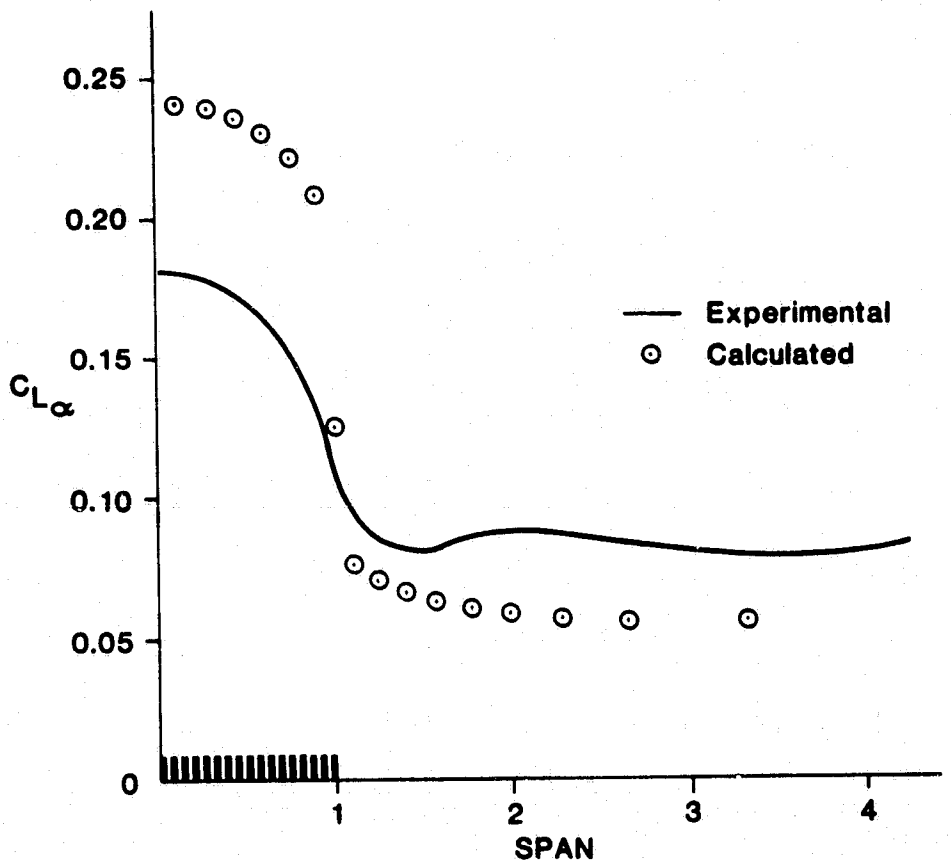


Figure 18. Continued

(b) Reference Figure V-5c, $V_{jet}/V_\infty = 2.5$

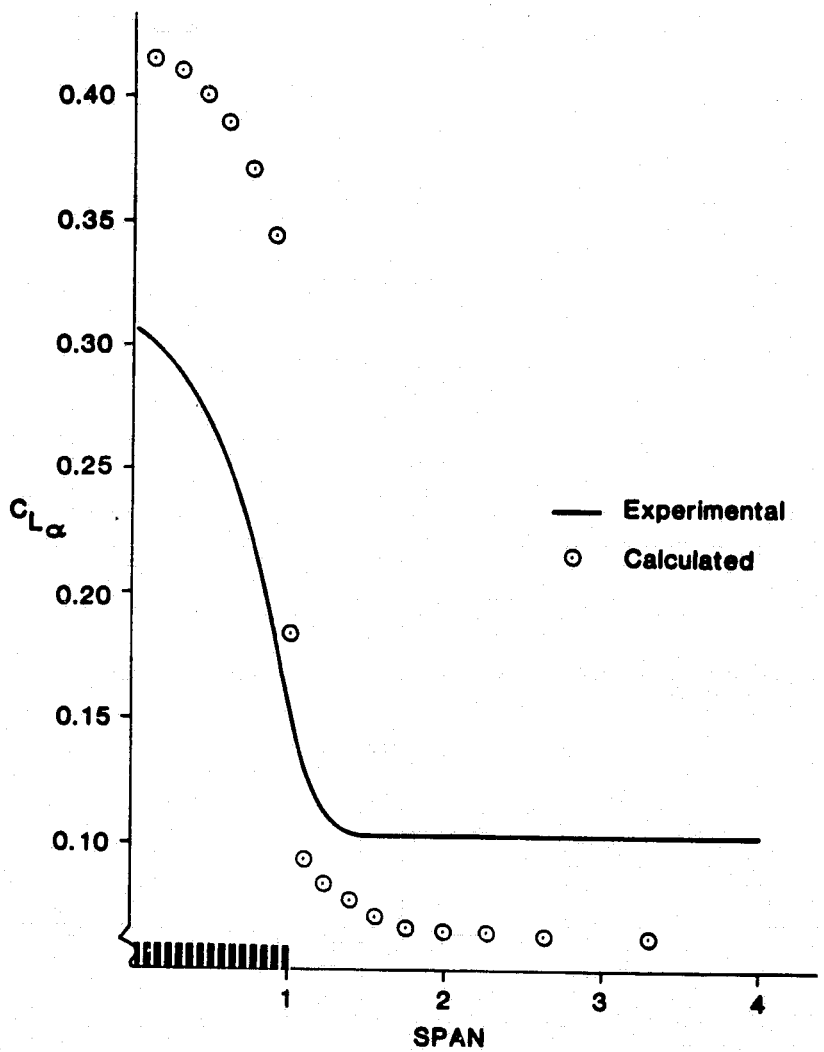


Figure 18. Concluded

(c) Reference Figure V-5b, V_{jet}/V_∞ = 3.3

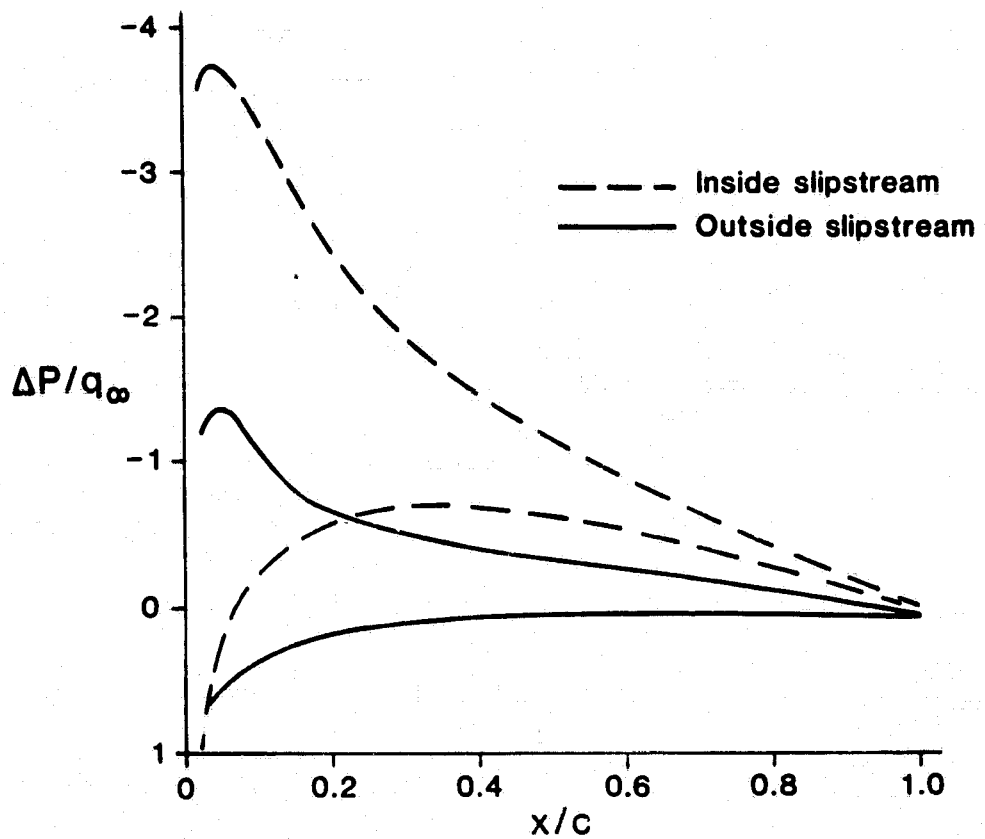


Figure 19. Experimental chordwise pressure distributions from Reference (7). Reference Figure V-1a, V_{jet}/V_{∞} = 2.5

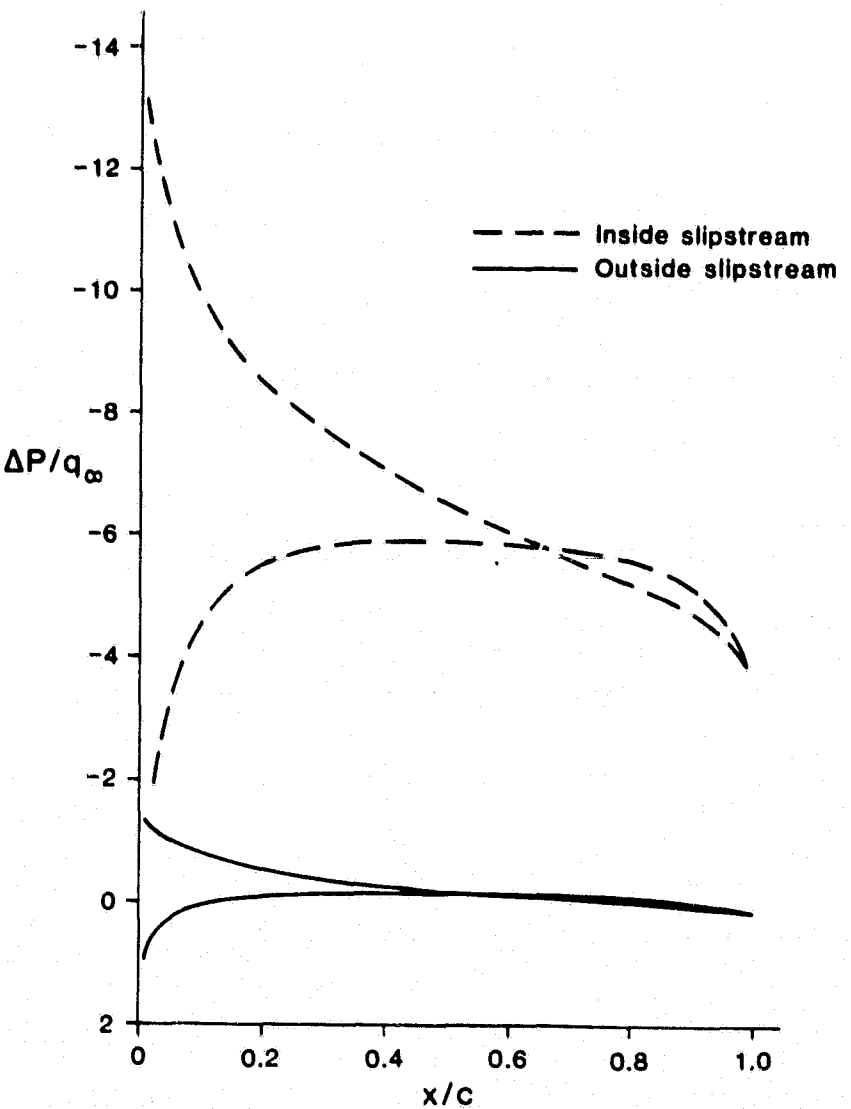


Figure 20. Calculated chordwise pressure distributions, $V_{jet}/V_{\infty} = 2.5$

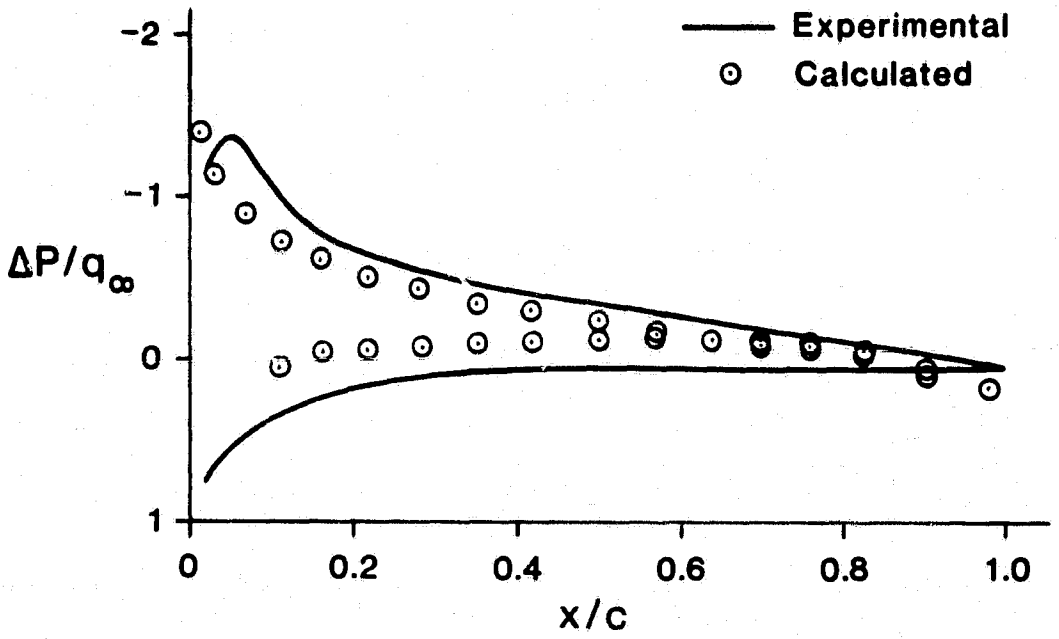


Figure 21. Comparison with Reference (7), calculated versus experimental.